

TURING

图灵系统与网络管理技术丛书

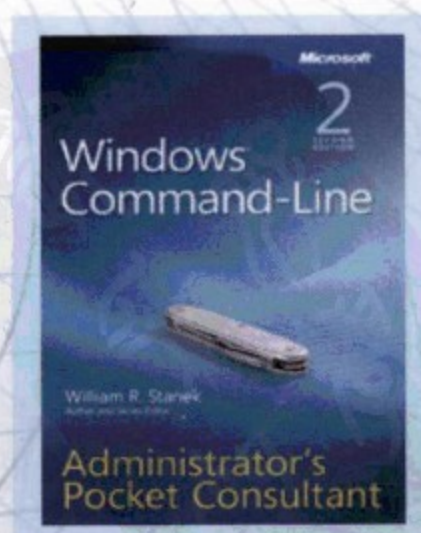
Microsoft

Windows Command-Line
Administrator's Pocket Consultant Second Edition

Windows 命令行详解手册 (第2版)

[美] William R. Stanek 著
王景新 等译

- Amazon五星图书，世界著名微软技术专家力作
- 涵盖Windows Server 2008与Windows Vista
- 信息密集，方便查询，让你轻松掌握Windows命令行的奥秘



人民邮电出版社
POSTS & TELECOM PRESS

Windows Command-Line
Administrator's Pocket Consultant Second Edition

Windows命令行详解手册 (第2版)

“本书堪称完美，我们公司人手一册……它提供了Windows系统管理员日常工作所需了解的一切，而且非常便于查阅……搞不定手上的任务时，你会立即发现它的价值。”

——Tarun Chachra, KSL公司CTO兼副总裁

“对于企业IT工程师来说，命令行是非常重要的管理手段。但是长期以来，国内一直缺乏很好的此类教材，本书则恰逢其时。”

——彭爱华（网名盆盆），微软高级讲师、
微软TechNet指定推荐博客ITECN创始人，

六届连任微软全球最有价值专家（Windows和虚拟化技术方向）

这是世界著名的微软技术专家William R. Stanek的一部力作，详细阐述了如何通过命令行来有效管理Windows操作系统，内容涵盖Windows Server 2008与Windows Vista。作者从大多数系统管理员的日常工作要求出发，将全书分为日志管理、磁盘管理、活动目录管理和网络管理等五个部分，细致入微地解释了几乎每一个命令和工具的适用场合及注意事项，并提供了数百个能立即用于实战的示例。读者既可以通过本书学习Windows命令行技术，也可以将其看作一本命令查询手册，随时放在手边，轻松应对日常工作。

本书适合Windows系统管理员、网络管理员及程序员阅读参考，同样也适合广大Windows用户学习使用。



William R. Stanek 世界知名的微软技术专家，微软MVP，拥有20多年系统管理和编程经验。他是一位广受赞誉的作家，已经累计撰写了100部著作，很多都是世界性的畅销书，已被翻译为四十多种文字。他也是经验丰富、深受欢迎的讲师。他的著作和培训课程已经影响了全世界数以百万计的程序员和管理员。

Microsoft

本书相关信息请访问：图灵网站 <http://www.turingbook.com>

读者/作者热线：(010)51095186

反馈/投稿/推荐信箱：contact@turingbook.com

分类建议 计算机/操作系统

人民邮电出版社网址：www.ptpress.com.cn



ISBN 978-7-115-21189-7



9 787115 211897 >

ISBN 978-7-115-21189-7/TP

定价：59.00元



图灵系统与网络管理技术丛书

Windows Command-Line
Administrator's Pocket Consultant Second Edition

Windows 命令行详解手册 (第2版)

人民邮电出版社
北京

图书在版编目(CIP)数据

Windows 命令行详解手册: 第2版 / (美) 斯坦尼克 (Stanek, W. R.) 著; 王景新等译. —北京: 人民邮电出版社, 2009.9

(图灵系统与网络管理技术丛书)

书名原文: Windows Command-Line Administrator's Pocket Consultant, Second Edition
ISBN 978-7-115-21189-7

I. W… II. ①斯…②王… III. 窗口软件, Windows—手册 IV. TP316.7-62

中国版本图书馆CIP数据核字(2009)第128686号

内 容 提 要

熟练使用命令行是系统管理员必备的技能之一, 本书从命令行的角度全面讲解了如何对 Windows 系统进行管理。书中首先概述了命令行的一些基本概念与技术, 之后将 Windows 系统管理任务进行分类, 并通过大量翔实的命令行实例分别讲解, 涵盖了 Windows 系统管理的主要工作。

本书适合于 Windows Server 2008、Windows Vista 系统管理员, 也可以供一般用户及命令行爱好者参考。

图灵系统与网络管理技术丛书

Windows命令行详解手册(第2版)

◆ 著 [美] William R. Stanek

译 王景新 等

责任编辑 傅志红

执行编辑 印星星

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京顺义振华印刷厂印刷

◆ 开本: 800×1000 1/16

印张: 26

字数: 714千字

2009年9月第1版

印数: 1-4 000册

2009年9月北京第1次印刷

著作权合同登记号 图字: 01-2008-3842号

ISBN 978-7-115-21189-7/TP

定价: 59.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

© 2009 by Microsoft Corporation. All rights reserved. Original English language edition © 2008 by William R. Stanek. All rights reserved. Published by arrangement with the original publisher, Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文简体字版由微软公司授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。



译者序

经过几个月的努力，本书的翻译工作顺利完成了。

据我所知，本书是讲解如何通过命令行管理Windows系统的不多见的书籍之一。市面上有一些讲解Windows管理的书籍，也有一些讲解命令行的书籍，但将二者结合起来，纯粹从命令行的角度来讲解如何完成复杂的Windows系统管理任务，这样的书籍非常少见。考虑到本书主要针对的是Windows Server 2008与Windows Vista，也可以说本书是全面系统地讲解通过命令行管理Windows Server 2008与Windows Vista的第一本技术书籍。对于Windows Server 2008与Windows Vista管理员而言，如果希望自己能更专业、更快捷地完成日常或特殊的管理任务，则本书是必不可少的一本参考指南。

本书最突出的特点是实用性。本书的撰写目标并不是成为一本关于Windows操作系统原理的专著，而是一本通过命令行进行Windows系统管理的操作指南。根据这一原则，本书没有那种似是而非的关于Windows操作系统原理的论述，而是在简单介绍了命令行的一些常识后，直接开始讲解如何通过命令行来完成特定的Windows管理任务，并给出了极其丰富的命令行实例，大部分这些实例我都进行过实际验证，可以正常运作，而这些实例执行的操作，比如计划任务、日志管理等，都是Windows系统管理员日常工作中不可或缺的重要内容。此外，本书在结构上分为几个部分，每一部分分别讲述不同的Windows管理主题，比如日志管理、磁盘管理、网络管理等，根据这种结构安排，管理员或读者可以非常便利地从相应部分中找到自己需要的命令行实例，更快捷地完成管理任务。

世间没有完美的事物，本书也一样，系统管理是一项高度复杂的任务，任何一本书也无法涉及每一个问题和每一个细节。本书虽然尽可能地给出了大量用于不同管理任务的命令行实例，但你并不能指望有这本书放在案头，就可以解决所有问题。不过，你可以通过这本书所体现的通过命令行解决系统管理问题的思路，通过本书给出的那些丰富的命令行实例，来学习和思考自己如何通过命令行来解决遇到的实际问题。很多时候，使用命令行要快捷得多，而且有些时候只能使用命令行，比如在Windows Server 2008的core-server安装模式下。

优秀的系统管理员不可能通过阅读一本书来造就，但一本优秀的书籍肯定会在造就优秀系统管理员的过程中起到重要的推动作用。如果本书能起到这种推动作用，那将是作者的荣幸，也将是译者的荣幸。

本书主要由王景新翻译。此外，参与本书翻译的还有叶俊、张乐锋、颜炯、富弘毅、何小威、奚丹、陈钢、王沛、陈小文、闫志强、薄建禄、林龙信、邓彬、焦贤龙、杨明军、马蓉、唐扬斌、刘志忠、田尊华、富弘毅、岳虹、肖国尊等。Be Flying工作室(http://blog.csdn.net/be_flying)负责人肖国尊负责本书翻译质量和进度的控制与管理，在此予以衷心感谢。译文虽经多次修改和校正，但是由于译者的水平有限，加之时间仓促，疏漏及缺点、错误在所难免，我们真诚地希望同行和读者不吝赐教，不胜感激之至。

是为序。

前言

本书设定的目标是为广大Windows管理员提供一本简洁而实用的工作指南，并具有良好的可读性。Windows管理员可以将本书放在案头，或者随身携带，随时方便查阅。本书详细讨论了管理员在使用命令行执行核心管理任务时可能遇到的几乎所有问题。由于本书的目标是用较小的篇幅提供尽可能多的信息，因此，读者无需在数百页篇幅的大量无关信息中苦苦搜索，而是可以直接精确找到满足自己工作需要的有益提示。

本书的编写目标是成为你在遇到任何Windows命令行管理问题时，都可以随时查阅的一本书。为达到这一目标，本书着重讲解管理员日常的管理程序、频繁执行的任务、清晰描述的示例，以及一些有代表性但并非一定包含的选项。总而言之，一方面保证内容的简洁，使得本书结构紧凑而便于查阅，另一方面要保证提供尽可能多的有用信息，成为管理员必备的宝贵资源。这样，本书既不是1000多页的长卷，也不是100来页的快速索引手册，而是一本颇有价值的参考指南，可以帮助你快速而容易地执行常见的任务，解决工作中遇到的实际问题，并执行一些高级的管理任务，比如自动监控、内存泄漏分析、磁盘分区、活动目录管理以及网络故障排除等一系列问题。

更多的在线内容。对本书有益的补充资料（或对原内容的更新）将被发布在Microsoft Press Online Windows Server and Client网站。随着Windows Server 2008最终版的成型，你会发现网上内容有对本书内容的更新、相关文章、到相关内容的链接、勘误信息、样章等资料。该网站网址为<http://microsoftpressro.Libredigital.com/Serverclient/>。

本书读者对象

本书讨论了Windows Server 2008与Windows Vista中的相关主题，面向的读者包括：

- Windows Server 2008管理员；
- 负责维护Windows Vista系统的技术支持人员；
- 承担部分系统管理职责的熟练用户；
- 所管理系统是从Windows Server以前版本升级到Windows Server 2008的管理员；
- 从管理其他操作系统平台转向Windows的管理员。

为包含尽可能多有用的信息，我只能假定读者对Windows有基本的理解、具备基本的网络技能，并且机器上已经安装了Windows系统。基于这一出发点，本书在结构上没有安排完整的章节来专门讲解Windows体系结构、Windows安装、Windows的启动与关机等内容，而主要是讲解任务调度、Windows系统监控、账号管理、网络服务管理以及更多高级主题。

此外，作者还假定你对Windows命令行与程序以及Windows用户界面相当熟悉，如果还需要进一步了解Windows基础知识，请参阅Windows文档。

本书组织结构

本书的设计宗旨是为Windows系统的日常管理任务提供一份有效的参考资料，因此，本书是以实际的管理任务来组织的，而不是以Windows的功能特性进行组织的。快速而便利的索引是这本必备指南的精华部分，本书的目录和索引有助于快速找到相关问题的解决方法。此外，本书还提供了很多其他的快速参考功能，包括快速的逐步操作指南、程序清单、表格（带有可用实例），还有很多丰富的交叉引用内容。在结构上，本书分为5部分共17章。

第一部分，“Windows命令行基础”，讲解了可通过命令行完成的基本管理任务。第1章概述了命令行管理工具、技术与概念。第2章在内容设计上旨在帮助读者充分利用shell命令的各种功能，其中详细讲解了如何使用不同参数组合来启动命令shell、如何控制命令路径的设置、存在哪些可用的重定向技术，以及如何使用多条命令组成的命令序列等。第3章讨论了创建命令行脚本的要素，包括如何设置变量、如何借助于条件控制进行工作，以及如何创建过程等内容。

Windows提供了很多命令行工具，帮助对系统的日常操作进行管理。第二部分，“使用命令行管理Windows系统”，讨论了可用于管理Windows系统的核心工具与技术。第4章探讨了用于配置Windows Server上的角色、角色服务以及功能的相关技术。第5章讨论了多种重要的管理工具，包括用于完成收集系统信息、Windows注册表操作、配置Windows服务以及远程关机等任务的工具。第6章讲述了Windows系统上的日志工具，可帮助识别与追踪系统中存在的各种问题，比如监控应用程序与服务、维护系统安全性，还能了解如何向系统日志与应用程序日志写入事件。在第7章，你将学习那些用于应用程序监控、进程检查以及性能维护的工具与技术。第8章讲解了可用于事件记录方式管理、企业级集中化事件记录以及性能数据的收集与报告生成等方面的技术。第9章讨论了任务自动化的方法与技术，以降低日常的工作压力。

接下来是本书的第三部分，“使用命令行管理Windows文件系统与磁盘”。用户需要使用硬盘来存储Word文档、电子表格等等类型的数据。只要你使用过Windows Vista或Windows Server 2008，就很可能使用过磁盘管理工具，磁盘管理工具对应的命令行是磁盘分区工具DiskPart，你可以使用DiskPart来处理大多数磁盘管理任务，也可以完成一些图形用户界面工具无法执行的任务。第10章介绍了DiskPart工具，也讨论了FSUtil、Chkdsk、CHKNTFS等工具。第11章讨论了基本的磁盘分区。第12章讲解了动态磁盘及其使用方法，也包括RAID的实施、管理与故障排解。

第四部分，“使用命令行管理Windows活动目录”。这一部分主要讲述用于活动目录配置、管理、故障排除的核心命令。第13章讨论了很多关键的目录服务管理工具，包括用于收集目录信息的工具等。第14章讲解了用于在活动目录中创建、管理计算机账号的工具，也包括如何将域控制器配置为全局编目与操作主机角色。第15章讲解了如何为活动目录中的用户与组创建、管理账号。

全书最后的第五部分，“使用命令行管理Windows网络”，讨论了网络打印、TCP/IP网络等相关主题。第16章主要讲解网络打印与打印服务。第17章讨论了如何使用命令行对TCP/IP网络进行配置、管理与故障排除。

附录A提供了对本书所讨论的命令行工具的快速索引；附录B提供了在使用网络服务shell（Netsh）时可用的上下文与命令的简明参考。通过Netsh，可以对本地主机与远程主机上多种网络服务的配置进行管理。

本书排版约定

为保证文本清晰易读，本书使用了多种元素标示某些内容。阅读本书你会发现，代码与命令清单

使用等宽字体印刷（那种讲述实际键入命令的情况除外，这些情况下命令是以**粗体**形式出现的）。在引入与定义新术语时，我将使用楷体。

其他一些约定如下所示。

- **注解**。为需要强调的某一问题给出详细解释。
- **最佳实践**。在使用高级的配置与管理概念时，给出可以使用的最佳技术与方法。
- **警告**。在可能存在需要注意的问题时，给读者警告信息。
- **更多信息**。为某一主题提供更多的信息。
- **真实场景**。在讨论高级主题时，提供真实场景下的建议。
- **安全告警**。指出重要的安全问题。
- **技巧**。提供有用的提示与额外的信息。

作者真诚希望，你可以通过本书尽可能快、尽可能高效地执行基本的管理任务。如果有什么想法，非常欢迎给我发邮件，我的邮箱是williamstanek@aol.com，谢谢！

支持信息

为保证本书的准确性，我们不遗余力。微软出版社通过网址：<http://www.microsoft.com/mspress/support>提供了本书的勘误信息。

如果你有评论、问题，或关于本书的任何想法，都可以通过如下的方法发送给微软出版社：
来信请寄：

Microsoft Press

Attn: Editor, Microsoft Windows Command Line Administrator's Pocket Consultant

One Microsoft Way

Redmond, WA 98052-6399

电子邮件：

mspinput@microsoft.com

需要注意的是，这些邮件地址不提供产品支持，要获取产品支持信息，可以访问微软的网站
<http://support.microsoft.com>。



致 谢

想要从根本上改变做某件事的方式，其困难程度超过了我的想象，但我仍然希望这种改变能给读者带来真正的价值。显而易见，有大量关于Windows管理的书籍，也有大量关于Windows脚本设计的书籍，但一直没有人写一本通过命令行管理Windows的书籍，这样一本书关注的应该是从命令行对Windows进行管理，而不是命令本身。我期待的是，辛苦撰写的这本书能有其独一无二的价值。本书不是那种一般意义的命令行书籍，那些书籍的内容通常是“这是某条命令，该命令可以完成某些任务，这是该命令的参数”。当然，本书也会包含类似内容（就像任何操作系统管理书籍一样），但是，本书讲解命令的语境是Windows系统的日常管理。本书将讲述如何执行日常的管理任务，并详细讲解如何通过命令行完成这些任务。因而，不管是希望学习如何管理日常操作、追踪Windows性能、查看事件日志、对磁盘进行分区、配置TCP/IP，还是执行数百种其他可能的系统管理任务，从本书中都会找到通过命令行解决问题的答案。

正如我在*Windows Server 2008 Administrator's Pocket Consultant* (Microsoft Press, 2008) 与*Internet Information Service (IIS) 7.0 Administrator's Pocket Consultant* (Microsoft Press, 2008) 两本书中所说的，微软出版社的编辑小组是一流的。本书在撰写过程中，得到了Karen Szall、Devon Musgrave、Maria Gargiulo以及其他人的大力帮助，在整个撰写过程中，他们都起到了重要的帮助作用，也非常感谢Martin DelRe的信任并促成本书的出版。

写完书稿只是整个出版过程的一部分工作，作者要耐心等待接下来的编辑与评审，但这个过程却可以保证读者手中书的质量。我必须承认，微软出版社的编辑与技术评审流程是最完善的——我曾经为多家出版社写过书。John Pierce是本书的责任编辑，在整个编辑过程中对我有很多帮助，这是我们第一次合作，非常愉快。Jim Johnson是本书的技术编辑，也是本书第一版的技术编辑，与他再次合作令人愉快。Becka McKay是本书的文字编辑。

我还要对Lucinda Rowley、Anne Hamilton、Chris Nelson表示感谢，他们在我写作生涯的很多时候都给予我帮助，在我最需要的时候，他们总是会适时出现。谢谢你们！

我很希望没有忘记感谢任何人，但如果忘记，则是无心之失。

资源分享网
PDG

目 录

第一部分 Windows 命令行基础

第 1 章 Windows 命令行概述	2
1.1 命令行基础	2
1.1.1 理解Windows命令shell	2
1.1.2 理解MS-DOS命令shell	5
1.1.3 理解Windows PowerShell	6
1.1.4 配置命令行属性	8
1.1.5 使用命令历史	9
1.2 使用补充的组件	10
1.2.1 在Windows Vista中使用微软远程服务器管理工具	10
1.2.2 注册远程服务器管理工具包	10
1.2.3 配置与选择远程服务器管理工具	11
1.2.4 删除远程服务器管理工具	11
1.2.5 删除远程服务器管理工具软件包	12
第 2 章 充分利用命令行	13
2.1 管理命令shell的启动方式	13
2.2 使用命令路径进行工作	15
2.2.1 管理命令路径	15
2.2.2 管理文件扩展与文件关联	16
2.3 标准输入、输出及错误日志的重定向	17
2.3.1 将标准输出重定向到其他命令	17
2.3.2 I/O与文件的重定向	18
2.3.3 标准错误输出的重定向	18
2.4 命令的结链与分组	19
2.4.1 使用命令链	19
2.4.2 命令分组	20

第 3 章 命令行脚本基础	21
3.1 创建命令行脚本	21
3.2 脚本的常见语句与命令	22
3.2.1 清除命令shell窗口	23
3.2.2 为脚本添加注释	23
3.2.3 管理文字的显示方式与命令回显方式	24
3.2.4 使用@对命令回显进行调整	25
3.2.5 设置控制台窗口的标题与颜色	25
3.3 向脚本传递参数	26
3.4 熟悉变量	27
3.5 在脚本中使用变量	28
3.5.1 变量命名	28
3.5.2 设置变量值	29
3.5.3 替换变量值	30
3.5.4 变量作用范围局部化	31
3.6 使用数学表达式	32
3.6.1 使用算术运算符与赋值运算符	32
3.6.2 理解运算符的优先级	33
3.6.3 模拟指数操作	33
3.7 命令行选择语句	34
3.7.1 使用if语句	34
3.7.2 使用if not语句	35
3.7.3 使用if defined与if not defined语句	35
3.7.4 使用嵌套的if语句	35
3.7.5 在if语句中进行比较	36
3.8 命令行迭代语句	36
3.8.1 迭代的基础	36
3.8.2 遍历一系列值	37

3.8.3 在成组的文件中迭代执行.....	38	第 6 章 事件记录、追踪与监控.....	79
3.8.4 在目录中迭代执行	38	6.1 Windows事件日志	79
3.8.5 分析文件的内容与输出	40	6.2 查看与过滤事件日志.....	82
3.9 创建子程序与过程.....	41	6.2.1 查看事件	82
3.9.1 使用子程序	42	6.2.2 过滤事件	83
3.9.2 使用过程	43	6.3 向事件日志中写入自定义事件.....	85
第二部分 使用命令行管理 Windows 系统		6.4 创建与使用保存的查询.....	86
第 4 章 部署 Windows 服务器.....	46	6.5 性能监控: 基础	89
4.1 服务器配置管理	46	6.5.1 理解如何在命令行中进行性能	
4.2 使用角色、角色服务与功能.....	47	监控	89
4.3 管理角色、角色服务与功能.....	51	6.5.2 追踪性能数据	90
4.3.1 ServerManagerCmd基础	51	第 7 章 进程监控与性能维护	94
4.3.2 查询已安装的角色、角色服务		7.1 管理应用程序、进程与性能.....	94
与功能	56	7.1.1 理解系统与用户进程	94
4.3.3 安装角色、角色服务与功能.....	57	7.1.2 检查运行中进程	95
4.3.4 移除角色、角色服务与功能.....	58	7.1.3 监控系统资源使用情况与进程.....	101
第 5 章 管理 Windows 系统	59	7.1.4 终止进程	106
5.1 检查系统信息	59	7.2 通过监控来检测与解决性能问题	108
5.2 操作注册表	61	7.2.1 监控内存分页与磁盘页面	108
5.2.1 理解注册表与键值	61	7.2.2 监控单个进程的内存使用与	
5.2.2 查询注册表值	63	Working Memory Set	109
5.2.3 比较注册表值	63	7.2.3 解决性能瓶颈	111
5.2.4 注册表键的保存与恢复	64	第 8 章 管理事件与性能日志	114
5.2.5 添加注册表键	65	8.1 管理事件日志	114
5.2.6 复制注册表键	65	8.1.1 开始使用Wevtutil	114
5.2.7 删除注册表键	66	8.1.2 列出可用的日志与已注册的事	
5.2.8 导入与导出注册表键	66	件发布者	115
5.2.9 加载与卸载注册表键	67	8.1.3 查看与改变日志配置	117
5.3 管理系统服务	69	8.1.4 导出与操作事件日志	119
5.3.1 查看已配置的服务	69	8.1.5 清除事件日志	122
5.3.2 启动、终止与暂停服务	71	8.2 企业级集中化事件记录机制.....	122
5.3.3 配置服务的启动方式	72	8.2.1 配置事件转发与收集	123
5.3.4 配置服务的登录方式	72	8.2.2 创建订阅	124
5.3.5 配置服务的恢复方式	73	8.2.3 管理订阅	128
5.4 从命令行重启与关闭系统.....	75	8.3 性能日志	130
5.4.1 管理本地系统的重启与关闭.....	76	8.3.1 开始使用数据收集器集	130
5.4.2 管理远程系统的重启与关闭.....	76	8.3.2 操作数据收集器集	131
5.4.3 添加关机或重启原因与注释.....	77	8.3.3 收集性能计数器数据	133

8.3.4 配置性能计数器警报	136	10.2 安装与管理硬盘驱动器	172
8.3.5 查看数据收集器报告	139	10.2.1 安装与检查新驱动器	172
第 9 章 计划任务的自动运行	141	10.2.2 检查驱动器状态与配置	173
9.1 在本地与远程系统上执行计划任务	141	10.2.3 修改驱动器分区风格	174
9.1.1 计划任务简介	141	10.3 操作基本磁盘与动态磁盘	175
9.1.2 监控计划任务	145	10.3.1 理解基本磁盘与动态磁盘	176
9.2 使用任务计划程序计划任务	146	10.3.2 设置活动分区	177
9.2.1 创建基本任务	146	10.3.3 改变磁盘类型: 基本磁盘与 动态磁盘的互相转换	177
9.2.2 创建高级任务	148	10.4 磁盘维护	178
9.2.3 管理任务属性	150	10.4.1 使用FSUtil获取磁盘信息并 管理文件系统	178
9.2.4 激活与禁用任务	150	10.4.2 检查磁盘的错误与坏扇区	180
9.2.5 将任务复制到其他计算机	150	10.4.3 修正磁盘错误	183
9.2.6 立即运行任务	150	10.4.4 对系统启动时的自动检测进 行控制	184
9.2.7 移除不需要的任务	150	10.5 磁盘碎片整理	185
9.3 使用Schtasks设置任务计划	151	第 11 章 对基本磁盘进行分区	188
9.3.1 使用Schtasks/Create创建计划 任务	151	11.1 获取分区信息	188
9.3.2 创建由Windows事件触发的计 划任务	156	11.2 创建分区	189
9.3.3 使用Schtasks /Change改变计划 任务	157	11.2.1 在MBR磁盘上创建分区	189
9.3.4 使用Schtasks/Query查询已配置 的任务	159	11.2.2 在GPT磁盘上创建分区	190
9.3.5 使用XML配置文件创建任务	159	11.3 管理盘符与挂载点	192
9.3.6 使用Schtasks /Run立即运行 任务	163	11.3.1 分配驱动器盘符或挂载点	192
9.3.7 使用Schtasks /End终止运行 中的任务	163	11.3.2 改变驱动器盘符或挂载点	193
9.3.8 使用Schtasks/Delete删除任务	164	11.3.3 移除盘符或挂载点	193
第三部分 使用命令行管理 Windows 文件系统和磁盘		11.4 格式化分区	194
第 10 章 配置与维护磁盘	166	11.4.1 使用FORMAT	194
10.1 使用DiskPart	166	11.4.2 使用FILESYSTEMS	195
10.1.1 DiskPart基础	166	11.4.3 格式化: 一个实例	197
10.1.2 DiskPart: 一个实例	167	11.5 管理分区	198
10.1.3 理解焦点及其内涵	167	11.5.1 将分区或卷转换为NTFS	198
10.1.4 DiskPart命令与脚本	167	11.5.2 改变或删除卷标	200
10.1.5 DiskPart: 脚本实例	170	11.5.3 压缩分区或卷	200
		11.5.4 扩展分区或卷	201
		11.5.5 删除分区	202
		第 12 章 管理动态磁盘上的卷与 RAID	203
		12.1 获取卷信息与状态	203
		12.2 创建并管理简单卷	205

12.2.1	创建简单卷	205
12.2.2	扩展简单卷	206
12.2.3	将动态磁盘联机	206
12.2.4	删除卷	207
12.3	通过动态磁盘上的RAID提供容错功能	207
12.3.1	实现RAID-0: 磁盘分割	208
12.3.2	实现RAID-1: 磁盘镜像与双控	209
12.3.3	实现RAID-5: 带奇偶校验的磁盘分割	210
12.4	管理RAID并从失效中恢复	212
12.4.1	分离镜像集	212
12.4.2	重新同步与修复镜像集	212
12.4.3	修复不带奇偶校验信息的RAID-0条带集	213
12.4.4	重建带奇偶校验信息的RAID-5条带集	213

第四部分 使用命令行管理 Windows 活动目录

第 13 章	核心目录服务管理	216
13.1	从命令行控制活动目录	216
13.1.1	理解域、容器与对象	216
13.1.2	理解活动目录中的逻辑结构与物理结构	217
13.1.3	理解区分名	218
13.1.4	使用活动目录命令行工具	218
13.2	使用DSQUERY命令进行目录查询	219
13.2.1	DSQUERY子命令及语法	220
13.2.2	使用名称、描述、SAM账号名进行搜索	221
13.2.3	设定搜索的登录域与Run As 许可权限	222
13.2.4	设定开始节点、搜索范围与对象限制	223
13.2.5	设定名的输出格式	225
13.2.6	结合使用DSQUERY与其他活动目录命令行工具	226

13.3	搜索问题用户与计算机账号	226
13.4	对象的重命名与移动	227
13.5	从活动目录中移除对象	228

第 14 章 管理计算机账号与域控制器

14.1	从命令行管理计算机账号概览	229
14.2	在活动目录域内创建计算机账号	230
14.2.1	创建计算机账号	230
14.2.2	定制计算机账号属性与组成 员关系	231
14.3	管理计算机账号属性	232
14.3.1	查看与寻找计算机账号	232
14.3.2	设置或修改计算机的位置与 描述信息属性	234
14.3.3	禁用与激活计算机账号	234
14.3.4	重置锁定的计算机账号	235
14.3.5	将计算机账号添加到某域中	236
14.3.6	对计算机与计算机账号进行 重命名	237
14.3.7	移动计算机账号	238
14.3.8	删除计算机账号	238
14.4	操作域控制器	239
14.4.1	安装与降级域控制器	239
14.4.2	在活动目录中发现域控制器	239
14.5	指定全局编目服务器	240
14.5.1	发现全局编目服务器	240
14.5.2	添加或移除全局编目服务器	241
14.5.3	检查缓存与优先的全局编目 设置	241
14.6	指定操作主机	242
14.6.1	发现操作主机	243
14.6.2	使用命令行配置操作主机 角色	244
14.7	发现只读的域控制器	246

第 15 章 管理活动目录用户与组

15.1	从命令行中管理用户账号概览	247
15.2	添加用户账号	249
15.2.1	创建域用户账号	249
15.2.2	自定义域用户账号属性与组 成员关系	250

15.2.3	创建本地用户账号	252	16.2.5	打印机重命名	281
15.3	管理用户账号	253	16.2.6	删除打印机	282
15.3.1	查看与查找用户账号	253	16.3	管理网络连接打印机的TCP/IP端口	282
15.3.2	确定单独用户账号的组成员关系	254	16.3.1	为打印机创建与改变TCP/IP端口	282
15.3.3	设置或更改用户账号属性	255	16.3.2	列出打印机使用的TCP/IP端口相关的信息	283
15.3.4	禁用与激活用户账号	256	16.3.3	删除打印机使用的TCP/IP端口	284
15.3.5	重置过期的用户账号	256	16.4	配置打印机属性	284
15.3.6	控制与重置用户口令	257	16.4.1	添加注释与位置信息	285
15.3.7	移动用户账号	258	16.4.2	共享打印机	285
15.3.8	用户账号重命名	258	16.4.3	在活动目录中发布打印机	285
15.3.9	删除用户账号	259	16.4.4	设置分隔页并改变打印设备模式	286
15.4	从命令行管理组账号概览	259	16.4.5	打印任务的调度与优先级设置	286
15.5	添加组账号	260	16.4.6	配置缓冲池与其他高级打印机选项	287
15.5.1	创建安全组与分发组	261	16.5	解决缓存问题	288
15.5.2	创建本地组并为其分配成员	262	16.5.1	检查Print Spooler服务	288
15.6	管理组账号	263	16.5.2	修复损坏的缓冲池	289
15.6.1	查看与寻找组账号	263	16.6	管理打印队列与单个打印任务	289
15.6.2	确定组成员关系	264	16.6.1	查看队列中的任务	289
15.6.3	改变组类型或范围	265	16.6.2	打印机的暂停与恢复	290
15.6.4	添加、移除或替换组成员	265	16.6.3	清空打印队列	290
15.6.5	移动组账号	267	16.6.4	暂停、恢复与重启单个文档的打印	291
15.6.6	组账号重命名	267	16.6.5	移除文档并取消打印任务	291
15.6.7	删除组账号	268	16.7	备份与恢复打印服务器配置	292

第五部分 使用命令行管理网络

第 16 章 管理网络打印机与打印服务

16.1	获取打印机的支持信息与故障排除信息	270	16.6.5	移除文档并取消打印任务	291
16.1.1	在命令行中操作打印机	270	16.7	备份与恢复打印服务器配置	292
16.1.2	追踪打印驱动程序与打印机信息	271	16.7.1	备份打印服务器的配置	292
16.1.3	获取用于容量规划与故障排除的打印详细统计资料	274	16.7.2	恢复打印服务器的配置	293
16.2	管理打印机	278	16.7.3	迁移打印机与打印队列	294
16.2.1	安装物理连接的打印设备	279	第 17 章 TCP/IP 网络的配置、管理与故障排除	295	
16.2.2	安装网络连接的打印设备	280	17.1	使用网络服务Shell	295
16.2.3	列出计算机上配置的打印机	280	17.1.1	操作Netsh上下文	295
16.2.4	查看与设置默认打印机	281	17.1.2	操作远程计算机	297
			17.1.3	操作脚本文件	298

17.2 管理TCP/IP设置	299	17.3.4 检查分片、重组、错误消息 的详细信息	312
17.2.1 配置IPv4	299	17.3.5 检查当前的TCP与UDP连接	313
17.2.2 配置IPv6	304	17.4 排除TCP/IP网络故障	317
17.3 支持TCP/IP网络	307	17.4.1 查看诊断信息	317
17.3.1 获取并保存TCP/IP设置	307	17.4.2 诊断常规的计算机配置问题	318
17.3.2 检查IP地址与网络接口 配置	309	附录 A 基本命令行工具参考	330
17.3.3 操作TCP Internet控制与错误 消息	310	附录 B Netsh 快速参考	367

Part 1

第一部分

Windows 命令行基础

本部分内容

- 第1章 Windows 命令行概述
- 第2章 充分利用命令行
- 第3章 命令行脚本基础

命令行内置在Microsoft Windows操作系统中，可以通过命令shell窗口访问。每个版本的Windows都有内置的命令行，用于运行内置的命令、工具以及脚本。尽管命令行是强大而多功能的，但有些Windows管理员从来都不使用它。有些Windows管理员乐于使用图形界面的管理工具，就可能一直使用这些工具——只是做一些鼠标单击的操作。

然而，对精通系统特性的Windows管理员、熟练的技术支持人员以及高级用户而言，Windows命令行是无法回避的。如果知道如何正确使用命令行（包括不同的时间场合应该使用哪些命令行工具，以及如何借助这些命令行工具来有效地完成工作），就意味着系统会平稳地运转，而不是问题频发。如果你负责多个域或网络的管理，那么对这些日复一日的大量管理操作而言，学习一下如何借助命令行提高效率不仅仅是重要的，而且是必需的。

本章我将讲述命令行的一些基础知识，包括如何使用内置的命令、如何运行命令行工具，以及如何使用其他支持工具进行工作。通过本章的讲解你会发现，默认安装下，Windows Vista与Windows Server 2008比以前的版本包含了更多的命令行工具。事实上，以前只有安装了Windows Support Tool与Windows Server Resource Kit工具之后才可用的很多工具，现在在默认安装下就可以使用了。

真实场景 在阅读本章以及本书其余部分时，应该记住的一点是，本书是以Windows Vista与Windows Server 2008为描述对象的。除非特别说明，本书中讲述的技术同时适用于这两类系统。有些情况下，本书中讨论的技术也可以应用于其他Windows操作系统，尽管某些选项或功能会有一些变化。无论哪种情况，在实际使用之前，你应该对这些命令、选项以及脚本进行测试，最好是在开发或测试环境（与实际工作环境隔离）下进行。

1.1 命令行基础

Windows的每一个新版本都会对命令行做一些扩展与增强，这种持续的更新使得Windows命令行的性能与多功能性都得到了很大的改善。用现今的Windows版本可以使用命令行完成很多以前版本无法完成的任务。为帮助读者以最快、最具效率的方式来使用那些可用的命令行选项，下面将讨论命令shell的选项与配置，还包括一些使用命令历史机制的相关提示。

1.1.1 理解 Windows 命令 shell

在Windows系统中，最常用的命令行就是Windows自带的命令shell。Windows命令shell（cmd.exe）

支持32位、64位两种环境，提供了使用Windows命令行进行工作的基本平台。在32位的Windows版本中，可以在%SystemRoot%\System32目录下找到这个32位的可执行程序cmd.exe。在64位的Windows版本的%SystemRoot%\System32目录中提供了64位的cmd.exe，在%SystemRoot%\SysWow64目录中提供了32位的cmd.exe。此外，Windows中还包括其他一些命令行，比如MS-DOS命令shell（command.com）与Windows PowerShell（powershell.exe），本章后面将对其进行讨论。

注解 %SystemRoot%指代的是环境变量SystemRoot。Windows操作系统有很多环境变量，这些环境变量可用于指代用户特定的或系统特定的一些值。通常，我会使用标准的Windows语法格式%VariableName%来指代环境变量。

要启动命令shell，可以使用“开始”菜单中的“搜索”对话框。单击“开始”，在“搜索”对话框中输入cmd，之后按Enter键。或者，使用鼠标依次单击“开始”、“所有程序”、“附件”，之后选择“命令提示符”。

可以通过不同的方法初始化Windows命令shell的环境，比如，绕过cmd.exe的启动参数，或者使用custom启动文件（该文件存储在%SystemRoot%\System32目录下）。图1-1展示了一个命令shell窗口。默认情况下，命令行的宽度为80个字符，命令shell则可以展示25行文本。如果更多的文本需要在命令shell窗口中显示，或者在命令shell窗口已满时输入新的命令，此时命令shell窗口中将显示当前输入的文本，而以前的文本会被冲掉。在某条命令输出结果进行时，如果想暂时停止显示，可以按Ctrl+S键，之后可以按Ctrl+S键恢复显示，或者按Ctrl+C键终止执行。

注解 custom启动文件用于需要特殊配置的MS-DOS程序，这些文件名为Autoexec.nt与Config.nt，存储在%SystemRoot%\System32目录下。



图1-1 命令shell将是你要使用的主要的命令行窗口

在这一来自Windows Server 2008的图中，显示的文本是：

```
Microsoft Windows [版本 6.0.6001]
版权所有 (C) 2006 Microsoft Corporation. 保留所有权利。

C:\Users\willams>
```

这里，命令行中的命令提示符展示了当前的工作目录，默认情况下该目录为%UserProfile%，代表的是当前用户的文件目录。命令提示符后跟随着一个闪烁的光标，表明命令行处于交互模式下。在交互模式下，你可以直接在提示符后键入命令，并按Enter键执行该命令。比如，键入dir之后按Enter键，就会得到当前目录列表。

除交互模式外，命令提示符还有一种批处理模式，该模式用于执行一系列命令。在批处理模式下，命令提示符逐一地读入并执行每一条命令。典型情况下，批处理命令是从脚本文件中读取的，但也可以在命令提示符下输入，比如，在使用FOR命令处理一组文件中的每一个文件时就是如此。（关于批处理脚本、循环、命令控制等内容，将在第3章进行更细致的讲述）

在使用Windows命令行时，要注意所使用命令的出处。本地命令（由微软内置在操作系统中的命令）包括下面两种。

□ **内部命令。**存在于命令shell内部，不包括单独的可执行文件。

□ **外部命令。**有自己的可执行文件，通常存在于%SystemRoot%\System32目录下。

表1-1展示了命令shell（cmd.exe）的内部命令列表，其中对每条内部命令给出了简短的描述。

表1-1 命令shell（cmd.exe）内部命令的简明参考

命 令 名	描 述
assoc	显示或修改当前的文件扩展关联
break	设置调试中断
call	在一个脚本内调用程序或其他脚本
cd (chdir)	显示当前目录名或改变当前目录位置
cls	清理命令窗口并擦除屏幕缓冲区
color	设置命令shell窗口的文本与背景色
copy	将文件从一个位置复制到另外的位置，或者将多个文件连接在一起
date	显示或设置系统日期
del (erase)	删除指定的文件、多个文件或目录
dir	显示当前目录或指定目录中的子目录与文件列表
dpath	允许程序打开指定目录中的数据文件（就像在当前目录中一样）
echo	显示命令行的文本字符串，设置命令回显状态（on off）
endlocal	变量局部化结束
exit	退出命令shell
for	对一组文件中的每一文件运行指定的命令
ftype	显示当前的文件类型或修改文件类型（文件扩展关联中使用）
goto	将命令解释器直接跳转到批处理脚本中某个标记行
if	命令的条件执行
md (mkdir)	在当前目录或指定目录下创建子目录
mklink	为文件或目录创建符号链接或硬链接
move	将一个或多个文件从当前目录或指定源目录移动到指定的目标目录，也可以用于对目录进行重命名
path	显示或设置操作系统用于搜索可执行文件与脚本的命令路径

(续)

命 令 名	描 述
pause	中断批处理文件的处理过程（挂起），等待键盘输入
popd	弹出由PUSHHD 保存的目录，使其成为当前目录
prompt	为命令提示符设置文本
pushd	保存当前目录位置，之后跳转到指定的目录（可选）
rd (rmdir)	移除目录（也可以移除其子目录）
rem	在批处理脚本或Config.sys中设置标记
ren (rename)	对一个或多个文件进行重命名
set	显示当前的环境变量，或者为当前命令shell设置临时变量
setlocal	在批处理脚本中标记变量局部化的开始
shift	改变批处理脚本中可替换变量的位置
start	启动一个单独的窗口，以便运行指定的程序或命令
time	显示或设置系统时间
title	设置命令shell窗口的标题
type	显示文本文件的内容
verify	在将文件写入磁盘后，指令操作系统对其进行验证
vol	显示磁盘卷标与序列号

要了解任意内部命令（以及大多数外部命令）的语法格式，可以在提示符后键入命令名，其后跟随/?，如下所示：

```
copy /?
```

随着对命令行的使用，你会发现，外部命令要远多于内部命令，包括那些与内部命令（内置在命令行中）非常相似的外部命令。大多数这些相似的外部命令对相应的内部命令进行了扩展与增强。比如，外部命令XCOPY要比内部命令COPY具备更丰富的功能，XCOPY允许复制目录树与文件，并提供了更多的参数。再如，使用外部命令SETX，可以将环境变量的变化直接写入到Windows注册表，持久性地改变环境变量，而SET只是临时性地改变。

提示 SETX是默认情况下Windows Vista与Windows Server 2008中支持的很多命令中的一条，也可以使用SETX获取当前注册表键值并将其写入到文本文件。

除此之外，内部命令与外部命令之间的差别并不是特别重要。很多Windows实用工具都有命令行扩展（允许将参数从命令行传递给工具），使用起来实际上与外部命令类似。

1.1.2 理解 MS-DOS 命令 shell

MS-DOS命令shell（command.com）包含了16位的命令，用于MS-DOS子系统与其他子系统。与大多数早期的Windows发行版不同的是，64位版的Windows Vista与Windows Server 2008中不再包含MS-DOS命令shell。在32位版的Windows Vista与Windows Server 2008中，可以使用RUN命令来启动MS-DOS命令shell。单击“开始”，选择“运行”，之后在打开字段中输入**command**，就可以启动MS-DOS命令shell。或者也可以在其他命令行中，键入**command**之后按Enter键，也可以启动MS-DOS

命令shell。

提示 如果在cmd.exe中启动MS-DOS命令shell，则命令shell标题将变为Command Prompt-Command。不再使用command.com时，可以键入exit退出MS-DOS命令shell并回到Windows命令行。

可以通过几种方式初始化MS-DOS命令shell环境，比如，将启动参数传递给command.com，或者使用config.nt启动文件（位于%SystemRoot%\System32文件夹）。与cmd.exe类似，默认情况下，MS-DOS命令窗口也是每行80个字符宽、窗口最多容纳25行文本。启动MS-DOS命令shell时，标准的显示文本是：

```
Microsoft(R) Windows DOS
版权所有(C) Microsoft Corp 1990-2001.

C:\>
```

与Windows命令shell类似，MS-DOS命令shell也有交互模式与批处理模式，也有微软内置的一些本地命令，这些命令可以划分为如下两组。

- **内部配置命令。**用于配置MS-DOS子系统的命令（存在于启动文件或程序信息文件中，比如Config.nt与Autoexec.nt），配置命令包括：BUFFERS、COUNTRY、DEVICE、DEVICEHIGH、DOS、DOSONLY、DRIVEPARM、ECHOCONFIG、FCBS、FILES、INSTALL、LOADHIGH、LASTDRIVE、NTCMDPROMT、SHELL、STACKS、SWITCHES。
- **标准的外部命令。**可以在命令提示符下键入，可以放置在脚本中，有时还可以在启动文件中使用。标准的外部命令包括：APPEND、DEBUG、EDIT、EDLIN、EXE2BIN、EXPAND、FASTOPEN、FORCEDOS、GRAPHICS、LOADFIX、MEM、NLSFUNC、SETVER、SHARE，这些命令也可以在cmd.exe中运行。

在MS-DOS命令shell中执行其他命令时，这些命令将被传递给32位的命令shell并由其真正执行，这也是可以在MS-DOS命令shell中使用内部命令COPY的原因。

1.1.3 理解 Windows PowerShell

Windows PowerShell (powershell.exe) 是一个功能完备的命令shell，包含了一些内置的命令(cmdlets)、内置的程序设计功能以及标准的命令行工具。在Windows Server 2008中，Windows PowerShell作为一个组成部分包含在其中。在其他Windows操作系统中，Windows PowerShell可以作为一个独立的程序下载使用。为保证Windows PowerShell是最新的版本，可以检查微软下载网站。

安装了Windows PowerShell之后，就可以在“开始\所有程序”菜单中（或者系统内相关的可执行程序中）找到程序快捷方式。通过“开始”菜单中的“搜索”对话框，可以启动默认的Windows PowerShell版本，具体方法是：单击“开始”，在“搜索”对话框中键入powershell，之后按Enter键。另外一种方法是，在其他命令行中，键入powershell，之后按Enter键。在32位系统中，上述方法将启动32位版本的Windows PowerShell。在64位系统中，上述方法将启动64位版本的Windows PowerShell。默认情况下，两个版本的Windows PowerShell都存储在%SystemRoot%\System32\WindowsPowerShell\Version目录中，这里，Version是系统中安装的PowerShell版本，比如v1.0或v2.0。在64位版本的Windows系统中，为保证兼容性，在%SystemRoot%\SysWow64\WindowsPowerShell\Version目录中可以找到32位

的PowerShell。

启动Windows PowerShell后，会看到类似于如下的消息：

```
Windows PowerShell V2
版权所有(C) 2008 Microsoft Corporation。保留所有权利。

PS C:\Users\williams>
```

通过指定-Nologo参数，可以在启动PowerShell时禁止该消息，如下所示：

```
powershell -nologo
```

注解 不管以哪种方式启动powershell，你都会知道自己确实在使用Windows PowerShell。因为启动后，命令提示符标题栏会改变为“Windows PowerShell”或“命令提示符-powershell”，并且当前路径是以PS引导的。

通过使用-Noprofile参数，可以启动Windows PowerShell而不加载profiles，使用如下命令：

```
powershell -noprofile
```

典型情况下，第一次启动Windows PowerShell时，你会看到一条消息，该消息声称脚本被禁用、列出的profiles没有被执行，这是Windows PowerShell默认的安全设置。为激活脚本的执行，可以在shell提示符下输入如下命令：

```
set-executionpolicy allsigned
```

这一命令将执行策略设置为所有脚本必须具有可信的签名才能执行。对约束条件没这样严格的环境，可以使用如下命令：

```
set-executionpolicy remotesigned
```

这一命令将执行策略设置为对从网站下载的脚本，只有具有可信源的签名才能执行，本地脚本则不需要数字签名即可运行。要想使得脚本不受约束地运行，则可以使用如下命令：

```
set-executionpolicy unrestricted
```

这一命令将执行策略设置为：允许脚本运行，而不管其是否具有数字签名。

使用Windows PowerShell时，可以在提示符下输入命令集的名，之后就会以与命令行命令类似的方式运行，也可以在脚本中执行cmdlets。在Windows PowerShell提示符下，键入help*-.*可以得到cmdlets变量的完整列表。

Cmdlets的命名采用的是动词-名词对的形式，其中，动词部分从通常的意义上表明该cmdlet的用途，名词部分则从具体的意义上表明该cmdlet针对的目标。比如，Get-Service这一cmdlet，既可以从所有的Windows服务中获取信息，也可以从某个专门指定的Windows服务获取信息。要了解某一特定cmdlet的帮助文档，可以键入help，其后跟随该cmdlet名，比如help get-service。

所有的cmdlets都有可配置的别名，别名可以充当执行的快捷方式。要列出所有可用的别名，可以在Windows PowerShell提示符下键入get-item -path alias。使用如下的语法格式，可以创建一个能调用任意命令的别名：

```
new-item -path alias:AliasName -value: FullCommandPath
```

这里，AliasName是创建的别名名称，FullCommandPath是运行命令的完整路径，比如：

```
new-item -path alias:sm -value: c:\windows\system32\ServerManagerCmd.exe
```

这一命令创建了别名sm，用于启动Windows Server 2008中的命令行管理工具Server Manager。要使用这一别名，你可以键入sm，其后跟随必需的参数，之后按Enter键。

1.1.4 配置命令行属性

如果你需要频繁地使用Windows命令shell，就一定希望对其属性进行个性化设置，使其更满足自己的需求。比如，你可以扩大缓冲区，以便看到原本在视野之外的文本，也可以对命令shell的大小进行定制、改变字体以及更多其他的内容。

要进行这些工作，首先启动命令shell，单击命令shell窗口顶部的命令提示符图标，或者鼠标右键单击控制台的显示条，之后选择“属性”。如图1-2所示，命令提示符属性对话框有下面4个选项卡。

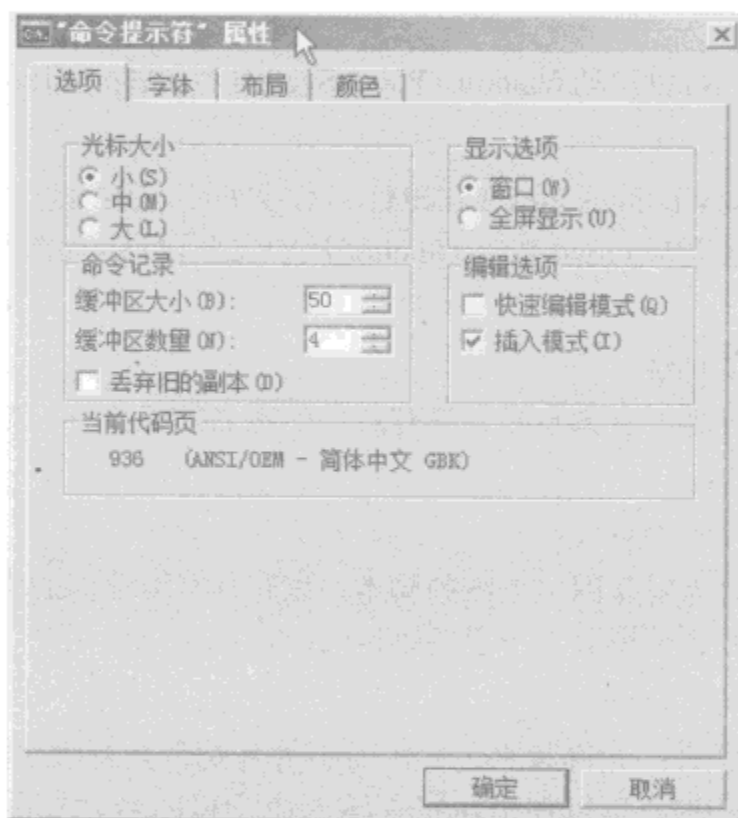


图1-2 为工作环境配置命令行属性

- 选项。在该项目中，可以设置光标大小、显示选项、编辑选项、命令历史等内容。如果想使用鼠标在命令窗口中剪切与粘贴文本，可以选择“快速编辑模式”。如果想以内容重写为默认的编辑模式，则应该清除“插入模式”。通过命令历史，可以设定以前使用过的命令在内存中的缓存模式。（在本章1.1.5节中，将会讲述关于命令历史的更多知识）

提示 工作在Windows Server 2008或Windows Vista SP1环境下，并且只使用文本形式的命令与工具时，一般需要使用全屏显示模式，以便降低命令提示符本身占据显示空间的比例。（要回到命令窗口模式，可以按Alt+Enter键）要回到Windows桌面，可以键入exit退出命令提示符。

- 字体。在该项目中，可以设置命令提示符所使用的字体大小与类型。光栅字体大小是根据像

素高度与宽度进行设置的。比如，8×12代表8个屏幕像素高、12个屏幕像素宽。其他字体大小是根据point大小设置的，比如10-point的Lucida Console。有趣的是，当选择point大小为n时，字体实际上是n个像素高。因此，10-point字体是10个屏幕像素高。这些字体也可以设定为粗体类型，此时将增加其屏幕像素宽度。

- **布局**。在该项目中，可以设置屏幕缓冲区大小、窗口大小以及窗口位置。通过增加缓冲区高度，可以方便地在前面的列表与脚本输出中进行滚动，较好的设置为1000~2000。通过增加窗口高度，可以一次查看命令shell窗口的更多部分，较好的设置为45行、12-point字体（800×600的屏幕）。如果想让命令提示符窗口处于某个特定的屏幕位置，则需要清除“由系统定位窗口”，之后指定一个位置，比如，要想让命令提示符窗口处于左上角，则需要将“左”、“上”均设置为0。
- **颜色**。在该项目中，可以设置命令提示符使用的文字颜色（使用“屏幕文字”）与背景颜色（使用“屏幕背景”）。“弹出窗口文字”与“弹出窗口背景”选项则分别对在命令提示符中运行命令过程中弹出的对话框的文字颜色、背景颜色进行控制。

命令shell属性更新完毕后，单击“确定”对所做设置进行保存。要注意的是，这些设置只对启动当前命令shell窗口的快捷方式有效，任何时候使用该快捷方式启动命令shell窗口时，这些设置都有效。然而，如果使用其他不同的快捷方式启动命令shell窗口，则需要将这些设置与其进行关联。

1.1.5 使用命令历史

命令历史缓存是Windows命令shell（cmd.exe）的一项功能，用于记录当前命令行会话中使用过的命令，允许在不需重新键入命令的情况下使用它。在前一节讨论的命令行属性对话框中，可以设置缓存命令数的最大值。默认情况下，该值为50。

通过如下步骤，可以改变命令历史缓冲区大小。

(1) 右击命令shell窗口的标题栏，选择“属性”，单击“选项”。

(2) 在“缓冲区大小”中，设置可以保存的命令数的最大值，之后单击“确定”保存所做的设置。

要注意的是，这些设置只对启动当前命令shell窗口的快捷方式有效，任何时候使用该快捷方式启动命令shell窗口时，这些设置都有效。然而，如果使用其他不同的快捷方式启动命令shell窗口，则需要将这些设置与其进行关联。

通过如下方式，可以访问记录缓冲区中保存的命令。

- **使用箭头浏览**。使用向上、向下箭头，在缓存命令列表中上下移动，发现需要使用的命令时，按Enter键，就可以按以前输入该命令时的方式执行该命令。也可以添加或修改命令参数，之后按Enter键，以便按当前需要执行该命令。
- **在命令历史弹出窗口中浏览**。按F7，可以显示一个弹出窗口，其中包含了缓存命令的列表。接下来，可以在其中使用箭头来选择命令。（另一种方法是按F9，之后通过在键盘上按相应的数字选择相应的命令，最后按Enter键执行。）之后按Enter键执行选中的命令，或者按Esc键关闭弹出的窗口而不执行命令。
- **搜索命令历史**。输入某条想要执行的命令的少数几个字符，按F8，之后命令shell会在命令历史中搜索第一条以键入的字符引导的命令，按Enter键即可执行该命令。或者再按F8，继续搜索命令历史中下一条匹配的命令。

使用命令历史时要记住的是，每一个cmd.exe实例有自己的命令缓存设置。因此，只有在相关联的

命令shell上下文中，命令历史缓存的相关设置才是有效的。

1.2 使用补充的组件

在设计Windows Server 2008与Windows Vista时，微软采用了可扩展的组件体系结构。借助于这种可扩展的体系结构，微软可以将新组件以安装包的形式提供给操作系统。典型情况下，微软下载站点会以微软更新独立程序包（.msu）文件的形式提供这些安装包。

新组件的安装与配置分为两个步骤：第一步是通过安装程序包来注册组件，第二步是使用适当的工具对组件进行配置。在Windows Server 2008上安装了新组件后，需要使用Server Manager中合适的向导来安装与配置新角色、角色服务或其他功能。在Windows Vista上安装了新组件后，则需要使用“Windows功能”对话框来安装与配置新的功能。在Windows Vista中，为激活远程管理功能，一般需要安装一个补充性组件微软远程服务器管理工具。

1.2.1 在 Windows Vista 中使用微软远程服务器管理工具

Windows Vista中的微软远程服务器管理工具（RSAT）是一个工具集，用于从Windows Vista主机上对Windows Server 2008的角色与功能进行远程管理。RSAT提供了对远程管理Windows Server 2008的支持（与Windows Server 2003管理工具包提供的功能类似），而不管Windows Server 2008本身是以Server Core形式安装的，还是以Full Server形式安装的。

RSAT以32位、64位两种版本提供给Windows Vista Business、Windows Vista Enterprise、Windows Vista Ultimate等操作系统，这里假定已经预先安装了Service Pack1（SP1）或后续补丁。如果运行的是32位版本的Windows Vista操作系统，则必须安装32位版本的RSAT。如果运行的是64位版本的Windows Vista操作系统，则必须安装64位版本的RSAT。你可以分别使用相应版本的RSAT来对32位、64位的Windows Server 2008进行管理。

需要注意的是，不应该在已经安装了Windows Server 2003管理工具包或Windows Server 2000管理工具包的机器上安装Windows Vista的微软远程服务器管理工具。在为Windows Vista安装微软远程服务器管理工具之前，必须删除以前安装的管理工具包。

由于远程管理工具的安装包是以软件更新的形式发布的，在微软知识库中，为其分配了一个识别码。记下这一数值，如果需要卸载或重新安装该安装程序包，该数值有助于对这一安装包进行定位。

1.2.2 注册远程服务器管理工具包

通过完成如下步骤，可以实现对Windows Vista的远程服务器管理工具安装包的注册。

(1) 获取微软远程服务器管理工具的版本信息，使其与将要安装该软件包的计算机的体系结构与Windows Vista系统服务补丁兼容。要获取适用于打了最新补丁的Windows Vista的远程管理工具软件包的最新版本，可以访问微软的下载网站（<http://download.microsoft.com>）。

(2) 在将安装包保存到本机或某网络共享位置处之后，就可以通过Windows资源管理器浏览到该位置。双击该安装程序，就可以开始安装过程。

(3) 当弹出对话框需要确认时，单击“确定”。阅读许可协议时，如果接受许可协议，单击“接受”继续，之后安装程序会以Windows Vista软件更新的形式安装该工具。

(4) 安装程序结束了对该工具的安装后,单击“关闭”。要注意的是,或许安装程序并没有明确提示要重启系统,但有时可能还是需要重启系统以完成安装过程。要确定是否需要重启,单击“开始”,观察“关闭计算机”按钮。如果该按钮是红色的,并且带有一个Windows更新的小图标,则需要重启来完成安装过程。

1.2.3 配置与选择远程服务器管理工具

只有在对安装包进行注册之后,远程管理工具才是可供选择并使用的。通过完成如下步骤,可以实现对将要使用的远程服务器管理工具的配置与选择。

(1) 依次单击“开始”、“控制面板”、“程序和功能”。

(2) 在“程序和功能”中,选择“打开或关闭Windows功能”。

(3) 在“Windows功能”对话框中,扩展“远程服务器管理工具”节点。

(4) 使用“功能管理工具”节点与“角色管理工具”节点下的选项,选择将要安装的远程管理工具,单击“确定”。这一类工具主要具备如下一些功能。

- Active Directory域服务工具。包括Active Directory域控制器工具与目录服务命令行工具。
- 分布式文件系统(DFS)工具。包括DFS管理单元,以及Dfsradmin、Dfscmd、Dfsdiag、Dfsutil等命令行工具。
- DNS服务器工具。包括DNS管理器管理单元与Dnscmd命令行工具。
- 故障转移集群工具。包括故障转移集群管理器管理单元与Cluster命令行工具。
- 文件服务器资源管理器工具。包括文件服务器资源管理器管理单元与Filescrn、Storrep等命令行工具。
- 网络负载均衡工具。包括网络负载均衡管理器管理单元与Nlb、Wlbs等命令行工具。
- SAN存储管理器工具。包括SAN存储管理器管理单元与ProvisionStorage命令行工具。
- Windows系统资源管理器工具。包括Windows系统资源管理器管理单元与Wsrn命令行工具。

选择了将要使用的工具后,Windows Vista会自动对其进行配置。在命令行,可以使用Windows Vista已经配置好的任意命令行工具。图形界面版本的工具则可以在管理工具菜单中选择与使用。

提示 如果管理工具菜单尚未出现在“开始\所有程序”中,可以通过如下的方法来将其在该处显示,以便快速便利地对远程管理工具进行访问。右击“开始”按钮,选择“属性”来显示“任务栏”与“开始菜单属性”对话框。在“开始菜单”选项卡上,单击“开始菜单”选项右侧的“自定义”,在“自定义「开始」菜单”对话框中,向下滚动直至发现“系统管理工具”选项,之后选择合适的选项,比如,“在‘所有程序’菜单与「开始」菜单中显示”,之后两次单击“确定”。

1.2.4 删除远程服务器管理工具

当不再需要使用远程服务器管理工具时,通过完成如下步骤,可以删除该工具的实现。

(1) 依次单击“开始”、“控制面板”、“程序”。

(2) 在“程序和功能”中,选择“打开或关闭Windows功能”。

(3) 在“Windows功能”对话框中,扩展远程服务器管理工具。

(4) 使用“功能管理工具”节点与“角色管理工具”节点下的选项，对想要删除的任意远程管理工具，清除对其进行的选定，之后单击“确定”。

1.2.5 删除远程服务器管理工具软件包

如果不再需要使用某台计算机进行远程管理，并且需要完全删除其上安装的远程管理工具时，通过完成如下步骤，可以完全删除远程管理工具的安装程序包。

- (1) 依次单击“开始”、“控制面板”、“程序”。
- (2) 在“程序和功能”中，选择“查看已安装的更新”。
- (3) 选择用于安装远程管理工具的更新程序，单击“卸载”。
- (4) 弹出提示时，单击“是”。



命令 shell提供了一个功能强大的环境，使得管理员可以使用命令与脚本完成很多工作。

如第1章中所述，可以在命令行中运行很多类型的命令，包括系统内置的命令、Windows工具以及带有命令行扩展的应用程序。每一个运行的命令都遵循同样的语法规则，而不论其出处。这些规则要求命令行中运行的每一个命令都由命令名开始，其后跟随着必需的或可选的参数。在参数中还可以使用重定向方法指定输入源、输出目标以及错误日志。

在命令shell中执行命令时，实际上涉及如下一些活动。

- (1) 命令shell使用实际值（实参）来替换用户在命令文本中输入的变量（形参）。
- (2) 在单一命令行中结成命令链（或分组）并传递的多条命令被分解为单独的命令，每条命令带有自己的命令名与参数，并按序分别进行处理。
- (3) 如果命令名带有文件路径，则命令shell将使用这一路径找到该命令，如果在指定位置找不到该命令，则命令shell将返回错误信息。
- (4) 如果命令名没有指定文件路径，则命令shell会尝试对该命令名进行内部解析。如果可以找到匹配项，则说明该命令是一条内置的命令，可以立即执行。如果找不到匹配项，则命令shell会在当前目录下查找该命令的可执行文件，之后搜索该可执行文件的命令路径。如果所有这些位置都无法找到该命令，则命令shell会返回错误信息。
- (5) 如果命令成功定位，则该命令会使用指定的参数运行，包括那些要求输入的参数。命令输出与错误信息将会回显到命令窗口，或者写入到存储输出信息与错误日志的位置。

从上面可以看出，很多因素都可以影响命令的执行过程，包括命令路径设置、使用的重定向技术，以及多条命令是否构成一个命令链或分组等。本章将根据这些因素对命令的执行过程进行分解，以便帮助读者最充分地利用命令行来高效完成自己的工作。不过在进行这些分解讨论之前，我们先来看一下启动命令shell的一些衡量要素，并介绍嵌套命令shell的概念。

2.1 管理命令 shell 的启动方式

在以前使用命令行工作时，你启动命令提示符的方式很可能是使用鼠标依次单击“启动”、“所有程序”、“附件”，之后选择“命令提示符”。然而，由这种方式启动的命令提示符只具备标准用户的权限，而不是管理员权限，因而无法执行很多管理任务。要以管理员权限启动命令提示符，则应该使用鼠标依次单击“启动”、“所有程序”、“附件”，之后用右键单击“命令提示符”，最后在运行方式中选择“以管理员身份运行”。

此外，还有其他一些方式也可以启动命令行。比如，使用开始菜单中的“搜索”对话框、“运

行”对话框，或者在命令shell窗口中键入cmd等方式。通过这些技术可以向命令行传递一些参数，包括用于控制命令行工作方式的参数（控制开关），以及用于执行其他命令的参数。比如，可以通过cmd /q命令以静默模式启动命令shell（这意味着命令回显被关闭），而如果希望命令shell在执行一条命令后就退出，就可以使用cmd /c命令，其后跟随着包含在引号中的命令文本。下面给出一个示例，启动命令shell，将ipconfig命令的输出发送到名为data的子目录下的一个文件中，之后退出命令shell：

```
cmd /c "ipconfig> c:\data\ipconfig.txt"
```

注解 要保证这一命令有效工作，data子目录必须存在。此外，从“开始”菜单的“搜索”对话框或“运行”对话框启动命令提示符时，命令提示符是以标准用户权限运行的。这意味着，你无法执行某些特定的管理任务，也不能将数据写入到对安全性比较敏感的系统位置。比如，如果你想把上面命令的输出重定向到C盘根目录（如c:\ipconfig.txt），则命令提示符没有足够的权限来创建该文件，从而导致命令执行失败。

表2-1总结了Windows命令shell（cmd.exe）的一些关键参数，要注意的是，有一些命令行参数是默认设置的。由于这一原因，命令行通常使用标准的ANSI字符编码进行命令输出（而不使用Unicode字符编码），并激活命令扩展，从而为大多数内置的命令扩充了功能。

表2-1 命令行的一些基本参数

参 数	描 述
/C	执行指定的命令，之后退出命令shell
/K	执行指定的命令，之后仍保持交互模式
/A	到文件（或管道）的命令输出设置为ANSI格式（默认设置）
/U	到文件（或管道）的命令输出设置为Unicode格式
/Q	开启静默模式，意味着命令回显被关闭。默认情况下，命令回显是开启的
/T:fg	为控制台窗口设置前台与背景颜色，这里fg是COLOR命令中定义的两个值
/E:ON	激活命令扩展（默认设置）
/E:OFF	禁用命令扩展

注解 有些参数不能与其他开关一起使用。比如，你不能同时激活ANSI与Unicode两种字符编码格式。如果同时使用/A与/U，或/E:ON与/E:OFF，则命令shell会选择命令行中最后传递的选项。

有些情况下，你可能需要为某命令行使用不同的环境设置与参数，之后返回到原始的设置，而不退出控制台窗口。为做到这一点，可以使用嵌套（nesting）技术。通过嵌套技术，你可以在命令行中启动另外的命令行，新启动的嵌套命令行会继承当前命令行的环境设置。在嵌套命令行中，你可以根据需要修改环境设置，并使用新的设置来执行命令与脚本。之后，在键入exit退出嵌套命令行时，会返回到原命令行，并且恢复原来的环境设置。

提示 在开始使用命令shell时，应该记住的是一些字符有特定的含义，命令shell在遇到这些字符时，会按照该字符特定含义所对应的流程执行。这些特殊字符包括<、>、(、)、&、|、@、^。

如果想把某个特殊字符用作常规字符，就必须使用相应的换码符来规避该字符原有的特殊含义，使得命令shell将其看成一个普通字符，而不再调用该字符特殊含义所对应的流程。这里使用的换码字符是^，在标准键盘上是按键6对应的字符，使用^作为引导，就可以规避字符的特殊含义。

2.2 使用命令路径进行工作

Windows操作系统使用命令路径来定位可执行文件，并根据文件的扩展名来判定文件类型是否为可执行文件。通过使用文件关联技术，可以将特定的应用程序映射为某种文件扩展名。接下来的两个小节将讨论使用命令路径、文件扩展以及文件关联的相关技术。

2.2.1 管理命令路径

通过PATH命令，可以查看可执行文件的当前命令路径。启动命令shell，在命令行中键入path，按Enter键。如果已经安装了Windows PowerShell，将会生成类似于如下的一些结果：

```
PATH=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;  
C:\Windows\System32\PowerShell\V2.0
```

注解 这里，分号(;)起到分隔不同路径的作用，命令shell使用分号来确定某文件路径的终点与另一文件路径的起点。

在登录系统后，命令路径是使用系统与用户环境变量进行设置的，即%PATH%变量。在设置时，路径中目录列出的顺序代表了命令行搜索可执行文件时采用的顺序。在上面的例子中，命令行会以如下顺序进行搜索：

- (1) C:\Windows\system32
- (2) C:\Windows
- (3) C:\Windows\System32\Wbem
- (4) C:\Windows\System32\PowerShell\v2.0

通过SETX命令，可以在系统环境中永久性地改变命令路径。比如，如果想将特定的目录用于脚本与应用程序，就可能需要更新路径信息。要做到这一点，可以使用SETX命令来向已存在的路径中添加一条特定的路径，比如使用命令setx PATH "%PATH%;C:\Scripts"。

注解 这里，引号与分号分别起到不同的作用。通过引号将%PATH%;C:\Scripts封装在一起，使其被命令shell理解为SETX的第二个参数。分号则与上面讨论的作用类似，用于确定某文件路径的终点与另一文件路径的起点。

提示 由于命令路径是在某次登录中设置的，必须先注销该次登录，之后重新登录，才可以在命令提示符中查看已修订的路径。如果不想注销，但又希望查看所做的命令路径设置是否正确，则可以使用“系统属性”对话框。方法是：在“控制面板\系统”中，单击任务面板中的“高级系统设置”，之后在“系统属性”对话框中的“高级”选项卡中单击“环境变量”。

在这个例子中，目录C:\Scripts被添加到现存的命令路径中。因而，如果此时向上面那样使用path命令，将会看到类似于如下的一些结果：

```
PATH=C:\Windows\system32;C:\Windows;C:\Windows\System32\wbem;  
C:\Windows\System32\PowerShell\V2.0;C:\Scripts
```

回想一下Windows使用的路径搜索顺序。由于路径是按顺序搜索的，因此，C:\Scripts目录将是最后一个被搜索的，这有时候会降低脚本的执行速度。要想让Windows能更快地搜索并执行脚本，可以将C:\Scripts设置为第一个搜索的目录。在这一例子中，可以通过如下命令进行设置：

```
setx PATH "C:\Scripts;%PATH%"
```

要注意的是，在设置命令路径时，有可能会无意间重写所有路径信息。比如，在设置路径时，如果没有指定环境变量%PATH%，就会删除所有其他的路径信息。一个可以确保重建命令路径的方法是将命令路径的副本写入到文件中。要将当前命令路径写入文件，可以在命令行中键入path>orig_path.txt。要注意的是，如果使用的是标准用户权限的命令提示符，而不是管理员权限的命令提示符，就无法将路径信息写入到对安全性敏感的系统位置。在这个例子中，你可以将路径信息写入到拥有访问权限的子目录中，也可以写入到自己的profile中。要将命令路径写入到命令shell窗口，则只需键入path。

现在，你可以看到一个列表或文件，其中包含了原始命令路径的列表。此外，path命令不仅可以列出当前的命令路径，还可以为当前命令shell设置临时的命令路径。比如，键入path %PATH%; C:\Scripts，就可以将C:\Scripts目录添加到当前命令shell的命令路径。

2.2.2 管理文件扩展与文件关联

通过使用文件扩展，在命令行中只需键入命令名就可以执行命令，有两种类型的文件扩展。

- 可执行文件的文件扩展。可执行文件是使用环境变量%PATHEXT%进行设置的，在命令行中键入set pathext，即可查看当前的设置。默认的设置是：PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC。根据这一设置，命令shell可以判定哪些文件是可执行文件，哪些不是，因而不需要在命令行中显式地指定文件扩展名。
- 应用程序的文件扩展。应用程序的文件扩展也就是文件关联（file association）。通过文件关联，可以将参数传递给可执行文件。这样，通过双击文件图标就可以打开文档、电子表格或其他应用程序文件。系统中每种已知的扩展都对应某种文件关联，通过键入assoc，其后跟随扩展名，就可以查看对应的文件关联，比如assoc.exe。反过来，每一文件关联都指定了某种文件扩展对应的文件类型，通过键入命令FTYPE，其后跟随文件关联，就可以查看对应的文件类型，比如ftype exefile。

对于可执行文件，文件扩展的顺序设定了命令行使用的搜索顺序（以每一个目录为基础）。因而，如果命令路径中某个目录下有多个可执行程序与提供的命令名匹配，那么.com文件将优先于.exe文件执行。

每一种已知的文件扩展（即便是可执行文件的）都有对应的文件关联与文件类型。大多数情况下，文件类型在形式上都是扩展文本（不包括句点）后面跟随关键字file，比如cmdfile、exefile或者batfile等。文件关联规定第一个传递的参数是命令名，其余参数应该传递给应用程序。

要查看已知的文件扩展对应的文件类型与文件关联，可以使用ASSOC命令与FTYPE命令。要查看文件关联，可以键入assoc，其后跟随文件扩展（包含句点）。ASSOC命令的输出是文件类型。所以，

如果键入 **ftype association** (这里 *association* 是 ASSOC 命令的输出), 你将会看到映射的文件类型。比如, 如果键入命令 **assoc .exe** (以便查看 .exe 可执行文件的文件关联), 之后键入 **ftype exe**。你会发现, 文件关联的设置为:

```
exefile= "%1" %*
```

根据这些信息, 在运行 .exe 文件时, Windows 会判断第一个值为要运行的命令, 其他的则为要传递的参数。

提示 文件关联与文件类型在 Windows 注册表中维护, 可以分别使用 ASSOC 命令与 FTYPE 命令进行设置。要创建文件关联, 可以键入 **assoc** 命令, 其后跟随扩展设置, 比如 **assoc .pl=perlfile**。要创建文件类型、设置文件类型映射以及如何使用与命令名一起提供的参数, 可以键入形如 **perlfile=C:\Perl\Bin\Perl.exe " %1" %*** 的命令。要了解关于文件关联与类型设置的更多信息, 可以在帮助与支持中心中查阅关于这两个命令的文档资料。

2.3 标准输入、输出及错误日志的重定向

默认情况下, 命令会从参数 (命令 shell 调用该命令时指定的) 中获取输入信息, 之后将输出信息 (包括错误信息) 发送到标准控制台窗口。然而, 有些时候, 你可能需要从其他信息源中获取输入信息, 或者将输出信息发送到某个文件或输出设备 (比如打印机), 也可能需要将错误信息重定向输出到文件, 而不是控制台窗口。要执行这些以及其他的重定向任务, 可以使用表 2-2 中介绍的技术, 接下来的几节对其中的技术进行了讨论。

表2-2 用于输入、输出、错误信息的重定向技术

重定向技术	描 述
command1>command2	将第一个命令的输出作为第二个命令的输入
command<[path]filename	从指定的文件路径中提取命令的输入信息
command>[path]filename	将输出发送到指定的文件, 必要的时候需要创建该文件或重写该文件 (如果已经存在)
command>>[path]filename	将输出附加到指定的文件 (如果该文件存在), 或者创建该文件并向其写入
command<[path]filename > [path]filename	从指定的文件中获取命令的输入, 之后将命令的输出发送到指定的文件
command<[path]filename>>[path]filename	从指定的文件中获取命令的输入, 之后将命令的输出附加到指定的文件
command2>[path]filename	创建指定的文件, 之后将错误输出信息发送到该文件。如果该文件存在, 则其内容会被重写
command2>&1	将错误输出信息发送到标准输出

2.3.1 将标准输出重定向到其他命令

大多数命令生成的输出信息都可以被重定向到其他命令 (作为输入信息)。要做到这一点, 需要

使用一种称为管道（**piping**）的技术，通过这种技术，一个命令的输出被用作下一个命令的输入。管道的通常语法是：

```
Command1 | Command2
```

这里，通过管道（**|**），**Command1**的输出被重定向为**Command2**的输入。也可以多次连续地进行这种重定向操作：

```
Command1 | Command2 | Command3
```

两个最常见的被用于管道操作的命令是**FIND**与**MORE**。**FIND**命令可以在文件或传递给命令的输入文本中搜索特定的字符串，如果找到匹配字符串，就列出匹配行的文本作为输出。比如，通过键入如下命令，就可以获取当前目录中所有.txt文件的列表：

```
dir | find /I ".txt"
```

MORE命令可以用于从其他命令接受输出信息来用作自己的输入，并且可以对接收的输出信息进行截断操作，使其可以在一个控制台页面中查看。比如，通过如下命令，就可以逐页阅读日志文件**Dailylog.txt**：

```
type c:\working\logs\dailylog.txt | more
```

要获取这两个命令语法的完整列表，可以在命令行中分别键入**find/?**与**more/?**。

2.3.2 I/O 与文件的重定向

另一种命令重定向技术是从文件中获取输入信息，这需要使用输入重定向符号（**<**）。比如，如下命令会对文件**Username.txt**中的内容进行排序，并在命令行中显示结果：

```
sort < usernames.txt
```

与从文件中读取输入信息类似，你也可以将输出信息发送到文件。要做到这一点，可以使用符号**>**来创建或重写指定的文件，也可以使用符号**>>**将数据添加到指定的文件。比如，通过如下命令，就可以将当前网络状态写入到文件**netstatus.txt**中：

```
netstat -a > netstatus.txt
```

遗憾的是，如果当前目录中已经存在一个同名的文件，上面的命令将重写该文件并创建一个新文件。如果想将这些信息附加到已有文件，而不是对其进行重写，则可以使用如下命令：

```
netstat -a >> netstatus.txt
```

输入、输出重定向技术也可以结合起来使用。比如，可以从某文件中获取输入信息，之后将命令的输出信息重定向到另外的文件。在这一实例中，从文件**usernames.txt**中获取用户名并对其进行排序，之后将排序后的用户名列表写入到文件**usernames-alphasort.txt**中：

```
sort < usernames.txt > usernames-alphasort.txt
```

2.3.3 标准错误输出的重定向

默认情况下，命令执行的错误信息会作为输出信息显示在命令行上。然而，在运行某些批处理脚本时，你可能会希望将错误信息输出到一个文件中，以便对错误进行追踪处理。有一种重定向标准错误的方法是让命令行将错误信息写入到标准输出。要做到这一点，可以键入重定向符号**2>&1**，

如下所示：

```
chkdsk /r > diskerrors.txt 2>&1
```

上例中，将标准输出与标准错误都发送到名为diskerrors.txt的文件中。如果只想保留错误的追踪信息，也可以只对标准错误进行重定向。通过如下命令，可以将标准错误发送到文件中，而标准输出则仍然在命令行中显示：

```
chkdsk /r 2> diskerrors.txt
```

2.4 命令的结链与分组

在前面部分中，讲述了包括管道命令的重定向技术。你可能想知道是否还有其他方法可以执行一系列命令，事实上确实还有其他方法。你可以将多条命令结成命令链并依序执行，根据前一条命令的成功或失败来条件性地执行后面的命令。也可以将多条命令分组，并条件性地执行。

在接下来的部分，你将学习命令结链与分组相关的技术。表2-3提供了关于命令结链、命令分组基本语法的快速索引。当然，该表不可能包含所有的这种语法，结链语法可以适当扩展，使得其他命令可以根据条件执行，命令分组的语法也可能会根据实际情况有所变化。

表2-3 命令结链、分组语法的快速索引

符 号	语 法	描 述
&	Command1 & Command2	执行命令1，之后执行命令2
&&	Command 1 && Command 2	如果命令1成功完成，则执行命令2
	Command 1 Command 2	只有命令1没有成功完成时才执行命令2
()	(Command 1 & Command 2) && (Command 3)	使用括号将命令分组，并根据成功与否条件执行
	(Command 1 & Command 2) (Command 3)	使用括号将命令分组，并根据失败与否条件执行

2.4.1 使用命令链

有时候，为提高效率，你可能希望以特定的顺序执行命令。比如，你可能希望进入某个目录，之后获取一份以日期排序的目录列表。通过使用结链技术，在命令行中输入如下一行命令即可完成这些任务：

```
cd c:\working\docs & dir /O:d
```

在脚本中，通常需要将命令结链，以便确保一系列命令如所期待的那样准确执行，当然，当后面命令的执行与否依赖于前面命令的成功或失败时，结链的作用就更为明显。下面的实例中，只要该日志文件存在，就移动它：

```
dir c:\working\logs\current.log && move current.log d:\history\logs
```

之所以采用这样一种方式，原因之一就是这种做法使得脚本不会输出错误信息。

有时候，你也可能希望只有在前一条命令失败时才执行某一任务。比如，在使用脚本向一组工作站发布文件时，其中的一些工作站对应的文件夹是C:\Working\Data，而另外一些工作站对应的文件夹是C:\Data。通过如下命令，你可以将一组文件复制到上述两种文件夹，而不需要关注工作站的具体配置：

```
cd C:\working\data || cd C:\data
xcopy n:\docs\*.*
```

2.4.2 命令分组

当需要执行多条命令时，可能需要将命令分组，以便防止冲突或确保命令以某种顺序执行。命令的分组是通过一组或多组括号实现的。要了解需要进行命令分组的原因，可以考虑这样一个例子。现要将主机名、IP配置以及网络状态等信息写入到某个文件，所以使用如下语句：

```
hostname & ipconfig & netstat -a > current_config.log
```

然而，检查日志文件时会发现，实际上只有网络状态信息被写入到文件中。之所以会有这种现象，是因为上述命令行中的命令会以如下顺序执行：

- (1) hostname
- (2) ipconfig
- (3) netstat -a>current_config.log

由于命令是依序执行的，因此，系统主机名与IP配置信息在命令行中输出，而只有网络状态信息被写入到日志文件。要将所有这些命令的输出写入到日志文件，可以采用如下方式对命令进行分组：

```
(hostname & ipconfig & netstat -a) > current_config.log
```

通过上面的方式，所有这3条命令的输出被收集在一起，作为一个整体重定向写入到日志文件中。你可以根据成功或失败对命令进行条件分组。比如，在下面的命令行中，只有在前两条命令都成功执行后，后一条命令才得以执行：

```
(cd C:\working\data & xcopy n:\docs\*.* ) && (hostname > n:\runninglog.txt)
```

在第3章，将会进一步讲述如何使用if与if...else结构对命令进行分组。



在图形用户界面占统治地位的IT世界，命令行脚本会提供哪些点选式对话框所不能提供的功能？实事求是地说，命令行脚本所能提供的功能要比大多数人所知道的多——对大部分把命令行脚本看成美化与增强版批处理文件（那种在8086处理器与MS-DOS环境中使用的文件）的人来说尤其如此。现今的命令行脚本环境是一个功能广泛的程序设计环境，包含如下一些要素：

- 变量；
- 算术表达式；
- 条件语句；
- 控制流语句；
- 过程。

通过这些程序设计要素，你可以做到：自动执行一些重复性的任务、在远离计算机时执行复杂的操作、发现他人错误放置的资源、执行很多其他耗时甚久的任务（通常需要在键盘键入数据才能完成的）等。命令行脚本不仅具备对命令行的完全的访问权限，也可以调用带有命令行扩展的任意工具。

3.1 创建命令行脚本

命令行脚本是包含待执行命令的文本文件，这些命令与通常情况下在Windows命令shell中键入的命令是一样的，将命令存储在命令行脚本中的好处是不再需要每次都键入命令，而是在需要的时候通过脚本很容易地执行命令。

由于脚本中包含的是标准的文本字符，因此可以使用标准的文本编辑器进行创建和编辑，比如记事本。在脚本中输入命令时，需要注意的是，每一条命令，或者每一组需要一起执行的命令，都应该保存在脚本的单独一行中，以便确保命令的正确执行。完成脚本创建后，使用.bat或.cmd文件扩展保存，这两种扩展对命令行脚本的处理与执行是一样的。比如，为创建一个脚本来显示系统名、Windows版本以及IP配置等信息，就可以在名为SysInfo.bat或SysInfo.cmd的脚本中输入如下3条命令：

```
hostname  
ver  
ipconfig -all
```

以上面的扩展名保存脚本后，就可以将其当作Windows工具一样执行：在命令shell中键入脚本名，之后按Enter键。命令shell会读入脚本文件，并逐一执行其中的命令，直至到达文件的尾部或遇到EXIT命令。对上面的示例脚本，执行后，命令行的输出信息与命令清单3-1类似。

命令清单3-1 示例脚本的输出

```

C:\>hostname
mailer1

C:\>ver
Microsoft Windows [版本6.0.6001]

C:\>ipconfig -all
Windows IP Configuration

Host Name . . . . . : mailer1
Primary Dns Suffix . . . . . : adatum.com
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : adatum.com


Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . . : 
Description . . . . . : Intel(R) PRO/100 VE Network
Connection
Physical Address. . . . . : X0-EF-D7-AB-E2-1E
DHCP Enabled. . . . . : No
Autoconfiguration Enabled. . . . : Yes
Link-local IPv6 Address . . . . . : fe80::2ca3:3d2e:3d46:fe99%9
(Preferred)
IPv4 Address. . . . . : 192.168.10.50
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.10.1
DNS Servers . . . . . : ::1
192.168.10.155
NetBIOS over Tcpip. . . . . : Enabled

```

观察上面的输出信息，你会发现，命令提示符与实际命令的显示方式与单独执行命令时的输出方式是一致的。之所以会出现这种情况，是因为在以默认的处理模式执行脚本时，命令shell做了一些附加的幕后工作。首先，命令shell显示命令提示符。之后，命令shell从脚本中读入一行，显示该行文本，并对其进行解释和执行。如果到达脚本文件的尾部，或者读入EXIT命令，则停止执行，否则就重复这一读入、显示和解释执行的过程。

在默认的处理模式中，命令回显是打开的，这有助于对脚本进行调试与故障排除。但对于经常使用的脚本，你可能并不需要这种显示模式。幸运的是，通过关闭命令回显，可以改变这种默认的显示模式，在3.2.3节将讲述这一问题。

3.2 脚本的常见语句与命令

到目前为止，本书主要讨论了一些命令及相关问题，但没有介绍什么是语句。尽管这些术语经常互换使用，但严格来讲，语句（statements）是指命令的关键字。比如`rem`语句，但有时候语句也可能是指一行代码，其中包含了该行上所有的命令文本。在有些程序设计语言中，如Java，每条语句必须以特定的字符（分号）作为终止符。在命令行脚本中，命令行并不寻找特定的终止符，而是寻找每行

的结束标志。比如，在命令解释器读入下面任意字符时就终止：

- 断行（比如按下Shift+Enter键）；
- 回车与断行（比如按Enter键）；
- 文件尾标志。

前面已经讨论了如何创建脚本，下面讨论脚本中可以使用的常用语句与命令。

- **Cls**。清除控制台窗口，重置屏幕缓冲。
- **Rem**。在脚本中创建注释。
- **Echo**。在命令行中显示消息、关闭或打开命令回显。
- **@**。以行为单位控制命令的回显方式。
- **Title**。设置命令shell窗口的标题栏。
- **Color**。设置命令shell窗口使用的文本色与背景色。

3.2.1 清除命令 shell 窗口

在将脚本输出写入到命令shell窗口之前，对其进行清空是一个较好的做法。要做到这一点，可以使用CLS命令。在命令行中键入**cls**，之后按Enter键。控制台窗口将清空，光标将定位到命令shell窗口的左上角，其后紧随着命令提示符，屏幕缓存中的所有其他文字也被清空。

在前面给出的示例脚本中，也可以加入CLS命令，如下所示：

```
cls
hostname
ver
ipconfig -all
```

3.2.2 为脚本添加注释

可以使用**rem**语句为脚本添加注释，为保证脚本的清晰易读，创建脚本时，应该为其添加如下一些注释信息：

- 脚本的创建时间与最后修改时间；
- 脚本的创建者；
- 脚本的用途；
- 脚本创建者的联系方式；
- 脚本输出是否保存及其保存位置。

通过上面这些脚本描述信息，不仅有助于其他系统管理员使用该脚本，也有助于脚本的创建者准确记忆脚本的用途。尤其在数个星期、乃至数月之后，创建者本人也很可能对脚本的用途已经陌生了。命令清单3-2中给出了一个脚本实例，其中包含了这些注释信息。

命令清单3-2 带有注释信息的示例脚本

```
rem *****
rem Script: SysInfo.bat
rem Creation Date: 2/28/2008
rem Last Modified: 3/15/2008
rem Author: William R. Stanek
rem E-mail : williamstanek@aol.com
```

```

rem *****
rem Description: Displays system configuration information
rem               including system name. IP configuration
rem               and Windows version.
rem *****
rem Files: Stores output in c:\data\current-sys.txt.
rem *****

hostname > c:\data\current-sys.txt
ver >> c:\data\current-sys.txt
ipconfig -all >> c:\data\current-sys.txt

```

在3.3节，将讲述如何把注释信息转化为自动帮助文档。现在，需要学习的是使用`rem`语句来完成如下任务。

- 在脚本中插入解释性的文字，比如，讲述某过程如何工作的文档。
- 防止命令的执行，在命令行，插入`rem`可以对命令进行注释和屏蔽。
- 隐藏某行命令的一部分，使其不再被解释执行。插入`rem`后，其后的部分将不再由命令shell进行解释和执行。

3.2.3 管理文字的显示方式与命令回显方式

ECHO命令有两种用途：一种是向输出（命令shell或文本文件）中写入信息，一种是打开或关闭命令回显。通常，在脚本中执行命令时，命令及其结果输出会在控制台窗口中显示，这称为命令回显（command echoing）。

要使用ECHO命令显示文字，可以输入`echo`，其后跟随要显示的文字，如下所示：

```

echo The system host name is:
hostname

```

要使用ECHO命令控制命令的回显方式，可以根据需要键入`echo off`或`echo on`，如下所示：

```

echo off
echo The system host name is:
hostname

```

使用ECHO命令，结合输出重定向技术，可以把输出信息发送到文件，而不是命令shell，如下：

```

echo off
echo The system host name is: > current.txt
hostname >> current.txt

```

要尝试关闭命令回显的情况，可以在命令shell中键入`echo off`，之后输入其他命令。你会发现，命令提示符将不再显示，而只能看到自己键入控制台窗口中的命令及其输出信息。与命令shell中类似，在脚本中使用ECHO OFF命令也可以关闭命令回显与命令提示符。为脚本添加ECHO OFF命令后，命令shell窗口或输出文件就不会被输入的那些命令本身所混淆，在只关注命令的输出信息时，这样做是有益的。

提示 顺便说一句，如果想确定命令回显状态是关闭还是打开，可以键入ECHO命令。如果命令回显是打开状态，会看到消息“ECHO处于打开状态”；如果命令回显是关闭状态，则会看到消息“ECHO处于关闭状态”。在脚本中使用ECHO OFF时，你可能会有一点疑问：如果说ECHO OFF的作用是关闭命令回显，那怎样防止ECHO OFF命令本身的回显呢？不用担心，3.2.4节会讲解这个问题。

真实场景 一些命令行程序设计者经常问我的一个问题是：如何在命令shell中回显一个空行？你可能认为在某行中键入ECHO命令即可，但实际上并非如此。就像上面的提示中所说的，键入echo将会显示命令回显状态。键入echo并在其后跟随空格也不能做到这一点，命令shell会认为这个空格是无意义的（在这个场景中），因此与只键入echo是一样的结果。要想让ECHO显示一个空行，必须键入echo和一个句点（echo.）。这里，句点是命令的一部分，必须紧随在ECHO命令之后。

3.2.4 使用@对命令回显进行调整

@命令可以以行为单位来防止命令回显到输出中，在一定意义上可以理解为特定于某行的echo off语句。使用@命令，可以通过如下方式关闭命令回显：

```
@echo The system host name is:
@hostname
```

使用@命令后，以如下形式输出命令提示符与命令的：

```
C:\>echo The system host name is:
The system host name is:
```

```
C:\>hostname
mailer1
```

会变成：

```
The system host name is:
mailer1
```

当然，@的真正价值在于可以使得命令shell不再显示命令提示符或ECHO OFF命令，确保脚本的输出信息只包含所键入命令的输出。下面给出一个脚本实例，其中使用@来隐藏ECHO OFF命令，使其不在输出信息中显示：

```
@echo off
echo The system host name is:
hostname
```

运行之后，上面脚本的输出为：

```
The system host name is:
mailer1
```

提示 我建议在所有命令行脚本前使用@ echo off。另外，在命令shell中键入@ echo off，也可以关闭命令提示符的回显。

3.2.5 设置控制台窗口的标题与颜色

编写命令行脚本时，可以加入一些特性使其更加醒目。前面已经讲述了一些基本的技术，包括使用ECHO OFF命令关闭命令回显、写入输出信息之前清空控制台窗口等。除这些工作外，在实际应用中，你可能还会希望设置命令shell窗口标题栏，或者改变窗口显示的颜色。

命令shell的标题栏定位在控制台窗口的上方。通常情况下，标题栏显示的是“命令提示符”，或

者命令提示符的路径。通过TITLE命令，可以对标题栏进行定制。该命令与ECHO命令类似，会在控制台的标题栏显示其后跟随的文字。比如，如果想将当前控制台窗口的标题设置为System Information，可以在命令行中输入如下命令：

```
title System Information
```

通过TITLE命令，不仅可以展示运行中的脚本名，还可以展示脚本运行的进度，如下所示：

```
rem add blocks of work commands
title Gathering Information
```

```
rem add blocks of logging commands
title Logging System Information
```

默认情况下，控制台窗口会以黑色背景展示白色文字。在第1章中曾经讲过，可以使用“命令行属性”对话框的“颜色”选项卡修改它。另外一种方法是使用COLOR命令，带一个由两个数字组成的十六进制代码参数，其中第一个数字代表背景颜色，第二个数字代表文字颜色。如下面的命令所示，该命令将文字颜色设置为蓝色，背景颜色设置为绿色：

```
color 21
```

表3-1中列出了COLOR命令可以使用的颜色代码。需要注意的是，文字颜色与背景颜色不能设置为相同，如果这样设置，则该命令不会起作用。此外，任何时候都可以使用COLOR命令（不带参数）恢复默认的颜色设置，如下所示：

```
color
```

表3-1 命令shell窗口的颜色代码

代 码	颜 色	代 码	颜 色
0	黑色	8	灰色
1	蓝色	9	淡蓝色
2	绿色	A	淡绿色
3	浅绿色	B	浅水绿色
4	红色	C	浅红色
5	紫色	D	淡紫色
6	黄色	E	淡黄色
7	白色	F	亮白色

3.3 向脚本传递参数

与大多数命令行工具类似，你也可以在脚本启动时向其传递参数。通过使用参数，可以设置脚本中特殊的变量，或者传递脚本运行需要的信息。参数应该跟随在脚本名之后，不同的参数应该使用空格分隔开（必要的时候还要将多个参数用引号封装起来）。在下面的实例中，向一个名为check-sys的脚本传递了参数mailer1与full：

```
check-sys mailer1 full
```

传递给脚本的每一个值都可以用形参进行检查与解释。脚本名本身由%0表示，%1代表传递给脚本的第一个参数，%2代表传递给脚本的第二个参数。依此类推，直到%9，代表传递给脚本的第9个参

数。比如，如果创建了`check-sys`脚本，并使用如下命令调用该脚本：

```
check-sys mailer1 full actual
```

则相关的参数值分别是：

- `%0-check-sys`;
- `%1-mailer1`;
- `%2-full`;
- `%3-actual`。

在脚本中，可以使用形参名来访问实参：`%0`代表脚本名，`%1`代表第一个实参，依此类推。比如，如果想要显示脚本名与传递给脚本的第一个实参，可以输入如下命令。

```
echo %0  
echo %1
```

如果向脚本传递了多于9个参数，多余的参数并不会丢掉，而是保存在一个特殊的变量`%*`中（百分号+星号）。`%*`代表了传递给脚本的所有实参，通过`SHIFT`命令即可查看多余的实参。如果不带参数运行`SHIFT`命令，则脚本形参左移一个。这意味着，`%0`代表的相关值被丢弃，并由`%1`代表的相关值替代，`%1`代表的相关值被`%2`代表的相关值替代，依此类推。必要的时候，也可以指定从何处开始进行形参移位，以便保留前面的参数。比如，如果使用下面的命令，则`%4`变为`%3`，`%5`变为`%4`，依此类推，但`%0`、`%1`、`%2`不受影响：

```
shift /3
```

3.4 熟悉变量

在命令行脚本中，我们通常所说的变量更可能指的是环境变量。环境变量有很多种来源，其中一些内置在操作系统之中，或者启动时来自系统硬件，这种变量称为内置的系统变量，对所有Windows进程都是可用的，而不管是否有人交互式地登录系统。系统变量也可以来自Windows注册表。除这种内置的系统变量之外，还有一些登录时设置的变量，这种变量称为内置的用户变量。内置的用户变量是相同的，而不管是哪个用户登录系统。此外，这些内置的用户变量只有在某次实际的登录会话（也即某用户登录了系统）中才是有效的。

通过在命令提示符中键入`set`，可以查看当前命令shell实例中所有已知的变量。除标准的系统与用户变量外，也可以在Windows运行中随时创建与设置变量——实际上这就是在命令shell中进行程序设计时所要去做的工作。使用`SET`命令与下面的语法，就可以为当前命令shell实例定义变量：

```
set variable_name=variable_value
```

比如：

```
set working=C:\Work\Data  
set value=5  
set string="Hello World"
```

有些变量（包括系统与用户环境变量）在命令shell中有特殊的含义，包括`path`、`computername`、`homedrive`以及很多其他重要的环境变量。此外，有一个需要学习和关注的环境变量`errorlevel`，该变量可以追踪最近使用命令的退出代码。如果命令正常执行，则错误级别为0；如果命令执行过程中出错，则错误级别会被设置为一个适当的非0值。下面是相关的错误类型值。

- 1。代表通常的错误。
- 2。代表执行错误，意味着命令没有正确执行。
- -2。代表算数错误，比如创建了一个命令shell无法处理的过大的数值。

可以以多种方式使用`errorlevel`变量，比如，检查特定的错误条件：

```
if "%ERRORLEVEL%"=="2" echo "An error occurred!"
```

或者，也可以使用如下的语法检查是否出现等于或大于指定的退出代码的错误条件：

```
if errorlevel 2 echo "An error occurred!"
```

注解 在本章3.5.3节与3.7节，将会更多地讲述`errorlevel`、`if`语句的相关内容。

使用完某变量之后，应该对其进行适当处理，以便释放该变量占用的内存，防止无意间再次引用该变量时出现问题与预期外的结果。要清除一个变量，很简单的方法是将该变量设置为空，如下所示：

```
set working=
```

之后，变量`working`将不复存在。

3.5 在脚本中使用变量

在脚本中，变量主要用于存储多种操作中涉及的各种值。与大多数程序设计语言不同的是，在脚本中声明变量时必须同时为其赋具体的值。这样做是有一定意义的，因为从实用主义的观点看，没有理由定义一个不包含任何值的变量。随后的几节将讨论在脚本中使用变量的一些关键概念，包括：

- 变量名；
- 变量值；
- 变量替换；
- 变量作用范围。

3.5.1 变量命名

命令shell可以区分变量名中使用字母的大小写，但在引用变量名时并不区分大小写。也就是说，变量名不是大小写敏感的，但是可以区分出大小写。除此之外，对变量名的限制是非常少的，你可以使用任意的字母、数字、字符组合在一起构成变量名。比如，下面的变量名，理论上都是有效的变量名：

```
2six  
85  
!  
?
```

这些变量名虽然有效，但易读性很差。因此，实际应用中一般不会采用这种变量名。在定义变量名时，最重要的一点就是变量名具有较好的描述性，即通过变量名本身就可以大概知道该变量的用途，如下所示：

```
System-name  
CurrentStats  
mergetotal
```



```
Net_Address
```

对使用或修改脚本的人而言，这种描述性很好的用户名是很有帮助的。当然，这种多字型变量名的命名方式有很多种，理论上都有效，不过大多数程序设计人员在创建这种多字型变量名时，第一个单词会以小写字母开头，而后续的其他单词会以大写字母开头，这是一种标准的命名约定。根据这种约定，上面列出的变量名实际上应该采用如下的形式：

```
systemName
currentStats
mergeTotal
netAddress
```

注解 要记住的是，命令shell不关心变量名是大写还是小写。也就是说，尽管命令shell有能力区分变量名中字母的大小写，但引用变量名时并不区分大小写。因此，对于上面定义的变量名 *systemName*，可以使用 *SYSTEMNAME*、*systemname*、甚至 *sYStemNAME* 等多种方式进行引用。

3.5.2 设置变量值

前面已经讲过，可以使用如下的语法来定义新变量。其中，*variable_name* 是变量名，*variable_value* 是变量值：

```
set variable_name=variable_value
```

在变量名与变量值中，空格都是有效的。因此，如果希望变量名与/或变量值中包含空格，可以在等号(=)附近设置空格。

与很多程序设计语言不同的是，命令shell不对不同的数据类型进行区分，所有变量都以字符串的形式存储，即便将变量值设置为数值时也是如此。因此，如下的变量值是以字符串形式存储的：

```
Current status:
311
"Error!"
12.75
```

要设置这些值，则需要使用如下命令：

```
set varA=Current status:
set varB=311
set varC="Error!"
set varD=12.75
```

要注意的是，有些字符是命令行的保留字符，包括@、<、>、&、|、^等字符。在使用这些字符时，不管出现在变量值中的哪个位置，都必须使用换码字符^对其进行换码（第2章中对其进行了讨论）。比如，要设置如下形式的字面意义字符串值：

```
2&3=5
2^3
```

就必须使用如下的变量值：

```
2^&3=5
2^^3
```

使用的命令形式则为：

```
set example1=2 ^& 3 = 5
set example2=2^^^3
```

注解 试图对上例中的变量值进行回显时，会发生一些奇怪的现象。与想要看到的等式不同，你或者会得到出错信息，或者会得到奇怪的值。回显变量值时之所以会出现这种现象，是因为命令shell对特殊字符进行了重复分析。如果想将变量值设置为包含特殊字符，同时又可以将该值向用户显示，就必须使用3个换码字符^。比如，对上面的例子，就需要使用`set example1=2^^^&3=5`或`set example2=2^^^3`。这样做是必要的，因为这些值被重复分析了（一次是在这些值被设置的时候，一次是在这些值被显示的时候）。

3.5.3 替换变量值

如果仅仅能使用SET命令对这些变量值进行设置，意义是很有限的。幸运的是，你可以通过其他方式访问变量。其中的一种是变量替换，用于对变量名与其真实值进行比较。在下面的命令行中，给出了一个这种替换的实例：

```
if "%ERRORLEVEL%"=="2" echo "An error occurred!"
```

该命令行的作用是确定环境变量`errorlevel`的值是否等于2，如果是，则显示一个字符串声明发生了错误。其中，用两个百分号将变量名封装在一起的作用是使得命令shell知道当前正在引用一个变量，如果没有使用百分号，则Windows会对"ERRORLEVEL"与"2"进行字面意义上的比较。另外还要注意其中使用了引号，其作用是确保对字符串值进行比较。

另外一种使用替换的方法是用变量的真实值替代变量名。比如，为创建一个可以在不同计算机上运行的脚本，就不能将系统根目录硬编码为C:\Windows，而是使用环境变量`systemroot`，在特定计算机上运行脚本时，该变量会引用该计算机的系统根目录。因此，在脚本中，应该使用如下代码：

```
cd %SYSTEMROOT%\System32
```

而不是使用如下代码：

```
cd C:\Windows\System32
```

为变量赋值时，也可以使用变量替换，比如：

```
systemPath=%SystemRoot%\System32
```

变量替换的作用是相当强大的，命令清单3-3中的代码展示了这一点。

命令清单3-3 示例脚本头

```
@echo off
@if not "%OS%"=="Windows_NT" goto :EXIT
@if "%1"=="*" (set INFO=echo && set SEXIT=I) else (set INFO=rem && set SEXIT=O)

%INFO% *****
%INFO% Script : SystemInfo. bat
%INFO% Creation Date: 2/28/2008
%INFO% Last Modified: 3/15/2008
%INFO% Author: William R. Stanek
%INFO% E-mail : williamstanek@aol.com
```

```

%INFO% *****
%INFO% Description: Displays system configuration information
%INFO%                including system name, IP configuration
%INFO%                and Windows version.
%INFO% *****
%INFO% Files: Stores output in c:\current-sys.txt.
%INFO% *****

@if "%SEEXIT%"=="I" goto :EXIT

@title "Configure Scheduling..."
cls
color 07

```

命令清单3-3是我在很多脚本中都会使用的标准的脚本头部信息，其中，第一个`if`语句用于检查当前运行的操作系统，如果是Windows 2000或后续版本，则脚本继续执行，否则会调用一个`goto`子过程。第二个`if`语句用于检查传递给脚本的第一个参数值，如果调用脚本时没有使用参数，则`%INFO%`实例会被`echo`（用于将脚本文档写入到输出）替换；如果调用脚本时使用了一个或多个参数，则`%INFO%`实例会被`rem`替代，用于指明相关联的行为注释信息。

注解 如果不能完全理解上面的实例，不必担心。在本章3.7节与3.9节，会详细讲述关于条件执行与子程序方面的内容。

3.5.4 变量作用范围局部化

在命令`shell`中，使用`set`命令对变量所做修改的作用范围是局部化的。这意味着，这些变量只适用于当前的命令`shell`实例，或当前命令`shell`中启动的命令`shell`（嵌套的命令`shell`），对其他系统进程则是无效的。进一步地说，退出命令`shell`（变量是在其中创建的）后，这些变量就不复存在。

有时候，可能需要对变量作用范围进行进一步的限制，使其作用范围局限于某一部分。要做到这一点，可以在脚本中创建一个局部范围，使得变量的改变只作用于脚本中某个特定的区域。在脚本的后面，可以终止局部范围，并将环境恢复到原始设置。

要完成局部范围的设定工作，可以在脚本中使用`SETLOCAL`命令来标记一个局部范围的开始，使用`ENDLOCAL`命令来标记一个局部范围的结束。使用这些命令时，实际上进行了一些幕后工作。调用`SETLOCAL`命令会创建当前环境的一个快照，在该局部范围内的任何变量变化都将局部化在该范围之内，调用`ENDLOCAL`命令时，局部范围内的变量变化与环境都将恢复至原始状态。下面给出了一个使用`SETLOCAL`命令与`ENDLOCAL`命令的实例：

```

@echo off
set sysCount=0
set deviceCount=0

rem Start localization
setlocal
set sysCount=5
set deviceCount=5
echo Local count: %sysCount% system edits ^& %deviceCount% dev ice checks
endlocal

```

```
echo Count: %sysCount% system edits ^& %deviceCount% device checks
```

该脚本的输出为:

```
Local count: 5 system edits & 5 device checks  
Count: 0 system edits & 0 device checks
```

可以看出,局部范围与嵌套命令shell是类似的,也可以嵌套多层局部范围。但嵌套层局部范围中的变化不会影响到上一层,尽管每一层都会从上一层中继承环境设置。

3.6 使用数学表达式

脚本中不时地需要进行一些数学运算,并将结果赋给某变量。与大多数程序设计语言类似,命令shell中也可以使用大量运算符来构成数学表达式,如下所示。

- **算术运算符**。用于执行标准的数学操作(比如加、减、乘、除)。
- **赋值运算符**。将赋值操作(由等号表示)与算数操作结合起来。
- **比较运算符**。用于对值进行比较,通常与if语句一起使用。
- **Bitwise运算符**。用于对二进制值序列进行操作。

算数操作是使用SET命令(带参数/A)进行的,比如:

```
set /a theTotal=18+2  
set /a theTotal=18*2  
set /a theTotal=18/2
```

所有的数学表达式都针对32位有符号整数进行运算,取值范围是 -2^{32} 到 $2^{32}+1$ 。如果超过了这一取值范围,就会报告算数错误(代码-2),而得不到正确的结果。

最常用的运算符是用于算数运算、赋值操作、比较操作的运算符,算术运算符与赋值运算符随后就会进行讨论,比较运算符则会在本章3.7.5节讲述。另外,还应该特别注意关于运算符优先级以及在脚本中模拟指数操作的讨论。

3.6.1 使用算术运算符与赋值运算符

算术运算符用于对数值进行一些基本的数学运算,这些数值或者可以表现为数字的形式,比如数字5。也可以表现为变量的形式(其中包含了要使用的变量值),比如%TOTAL%。

表3-2总结了可用的算数操作符与赋值运算符。大部分算术运算符的含义都是非常明显的,比如,运算符*用于乘法操作,运算符/用于除法操作,运算符+用于加法操作,运算符-用于减法操作。等号(=)表示赋值运算符,用于将变量值赋给变量,%(模数)则用于获取除法操作的余数部分。比如,用60除以8,则结果为7余4,使用%(模数)运算符时,4就是得到的结果。

下面给出了几个使用算术运算符的实例:

```
set /a theCount=5+3  
set /a theCount=%nServers% + %nWstations%  
set /a theCount=%nServers% - 1
```

提示 前面曾经讲过,变量中存储的变量值都是字符串,这一点在使用算术运算符的语境下仍然是对的。不过,命令shell可以判断出什么情况下字符串中包含的都是数字。因此,在算数表达式中使用变量是可以的。要记住的关键点是使用正确的语法进行替换操作: %variableName%。

表3-2 算数与赋值操作运算符

算数操作符	赋值操作符
+(加)	+(加之后赋值)
-(减)	-(减之后赋值)
*(乘)	*(乘之后赋值)
/(除)	/(除之后赋值)
%(模)	%(模之后赋值)

3

使用赋值运算符，可以进行递增、递减、按比例增加、缩减等操作。这些运算符实际上也结合了算术运算符的功能。比如，+=运算符用于对某个值进行加操作后再重新赋值，实际上就结合了+运算符与=运算符的功能。因此，如下两个表达式是等价的，返回的是同样的结果：

```
set /a total=total+1
set /a total+=1
```

3.6.2 理解运算符的优先级

使用算术运算符工作时，理解运算符优先级是必要的。在表达式中涉及到多个运算符时，运算符优先级会决定其执行顺序，比如：

```
set /a total=8+3*4
```

如果简单地从左至右进行运算，则上面表达式的结果为44（8+3=11，11*4=44）。然而，在数学运算中，上述表达式的结果实际上应该是20（3*4=12，8+12=20）。这是因为，表达式中的运算符具有如下的优先级：

- (1) 模数运算；
- (2) 乘法操作与除法操作；
- (3) 加法操作与减法操作。

注解 表达式中包含多个同优先级的运算符时，则采用自左至右的顺序进行运算。因此，set /a total=10-4+2的值为8（10-4=6，6+2=8）。

然而，在数学运算中，可以使用括号将算术表达式的某些部分包含在一起，从而改变运算的优先级。因此，通过如下表达式：

```
set /a total=(8+3)*4
```

可以使得命令shell将其解释为：8+3=11、11*4=44，所以结果为44。

3.6.3 模拟指数操作

尽管可以在命令行中进行很多数学运算，但无法使用指数操作运算符。不过，可以通过人工的方式进行指数运算。比如，要计算 2^3 的值，最简单的方法是输入如下命令：

```
set /a total=2*2*2
```

上述表达式的结果为8。同样地，要计算 10^5 的值，可以输入如下命令：


```
set /a total=10*10*10*10*10
```

上述表达式的结果是100,000。

3.7 命令行选择语句

前面讲解了如何使用变量与规范的表达式，下面讨论命令行脚本中一些更高级的功能：在命令行中使用选择语句。如果想控制脚本的执行流程，但相应的判别条件只能在运行时才可以确定，则可以采用如下方法。

- 使用if语句。在条件为真（比如，操作系统为Windows 2000及后续版本）时执行某语句，否则绕过该语句。
- 使用if not语句。在条件为假（比如，某系统中不包含C:\Windows目录）时执行某语句，否则绕过该语句。
- 使用if...else语句。在条件匹配（真或假）时执行某语句，否则执行另外的语句。

尽管本章前面的一些例子中已经使用了条件执行语句，但我们没有讨论这些语句的语法，以及相关关联的比较操作符。如果你不具备程序设计的背景知识和经验，你可能会为这些语句的强大功能与灵活性而深感惊讶。

3.7.1 使用 if 语句

if语句用于条件分支，可以将脚本的执行引导向两条不同的执行路径，其基本语法为：

```
if condition (statement1) [else (statement2)]
```

这里，每条语句可以是单一的命令，也可以由多条命令组成。这些命令可以组成命令链，可以使用管道连接，也可以使用圆括号进行分组。条件可以是任意的表达式，该表达式可以返回布尔类型的True（真）或False（假）。else子句是可选的，也就是说，你也可以使用如下的语法：

```
if condition (statement)
```

提示 理论上讲，圆括号并不是必需的，但使用圆括号可以使得语句的结构更加清晰，尤其在条件中包含了echo语句或带有参数的命令时更是如此。在这些情况下如果不使用圆括号，则当前行中在该语句之后的所有内容将被解释为语句的一部分，所以经常会导致错误。

if语句的工作方式是：如果条件为真，则执行语句1，否则执行语句2（前提是if语句中也使用了else子句）。在任何情况下，if子句与else子句都不会同时执行。参考下面的实例：

```
if "%1"=="1" (echo is one) else (echo is not one)
```

这里，如果传递给脚本的第一个参数是1，则is one被写入到输出。否则，is not one被写入到输出。

在每个条件后面，命令shell只会执行一个语句。典型情况下，该语句可以是单一的命令。如果想在判别条件后执行多条命令，可以使用命令管道、命令链、命令分组等技术，如下所示：

```
if "%1"=="1" (hostname & ver & ipconfig /all) else (netstat -a)
```

这里，如果第一个参数为1，则第一组圆括号中的所有3条命令都将执行。

3.7.2 使用 if not 语句

如果希望在条件为假 (false) 的情况下才执行命令, 则可以使用 *if not* 语句, 其基本语法为:

```
if not condition (statement1) [else (statement2)]
```

这里, 命令 *shell* 对条件进行判断。如果条件为假, 则执行语句1。否则不执行语句1, 且命令 *shell* 继续执行到语句2 (如果存在)。*else* 子句是可选的, 也就是说, 除上面的语法外, 也可以使用如下的语法:

```
if not condition (statement1)
```

参考下面的实例:

```
if not errorlevel 0 (echo An error has occurred!) & (goto :EXIT)
```

这里, 命令 *shell* 对非0的错误条件进行检查。如果没有错误发生 (也即错误级别为0), 则命令 *shell* 跳转到下一条语句。否则, 命令 *shell* 将 *An error has occurred!* 写入到输出, 并退出脚本。(本章后面将讲述关于 *goto* 语句与子过程的详细内容)

3.7.3 使用 if defined 与 if not defined 语句

最后两种可以使用的 *if* 语句是 *if defined* 与 *if not defined*, 这两种语句可用于检查某变量是否存在, 两种语句的语法格式分别为:

```
if defined variable statement
```

和

```
if not defined variable statement
```

在 *shell* 脚本中, 这两种语句都是有益的。第一种情况, 如果指定的变量存在, 则执行某条命令。第二种情况, 如果某变量不存在, 则执行某条命令。参考如下的实例:

```
if defined numServers (echo Servers: %numServers%)
```

这里, 如果变量 *numServers* 已经定义, 则脚本向输出写入信息。否则, 脚本跳转到下一条语句。

3.7.4 使用嵌套的 if 语句

嵌套的 *if* 语句是指在 *if* 语句中又包含了其他的 *if* 语句, 这种语句在程序设计中是很常见的, 命令 *shell* 程序设计中也是如此。使用嵌套的 *if* 语句时, 需要注意如下几点。

- 使用花括号定义代码块, 使用 @ 符号标记嵌套循环语句的开始。
- *else* 语句总是与同一代码块内最临近的 *if* 语句匹配, 前提是该 *if* 语句没有与任何其他的 *else* 语句关联起来。

下面给出一个实例:

```
if "%1"=="1"(  
@if "%2"=="2" (hostname & ver) else (ver)) else (hostname & ver &  
netstat -a)
```

这里, 第一个 *else* 语句是与语句 *if "%2"=="2"* 匹配的, 后一个 *else* 语句则与语句 *if "%1"=="1"* 匹配。

3.7.5 在 if 语句中进行比较

如前面一些例子中所示，用于控制 *if* 语句的表达式中经常涉及到比较运算符。最基本的字符串比较类型是使用等号 (=) 对两个字符串进行比较，如下：

```
if stringA==stringB statement
```

这里，对字符串进行了字面意义的比较。如果两个字符串完全等同，则执行后面的语句。这种语法格式对字面意义的字符串是有用的，但对脚本不是很适用。形参与实参中可以包含空格，或者某变量可以不进行赋值。在这种情况下，如果进行字面意义的比较，就会得到错误信息。为避免这种错误，可以使用双引号包含起来进行字符串比较，并防止大多数错误，比如：

```
if "%varA%"=="%varB%" statement
```

或者：

```
if "%varA%"=="string" statement
```

字符串比较总是区分大小写的，除非指定了 */i* 参数。*/i* 参数会使得命令 *shell* 在进行字符串比较时忽略字母的大小写，如下所示：

```
if /I "%1"=="a" (echo A) else (echo is not A)
```

要进行更高级的相同性测试，可以使用表3-3中列出的比较运算符，这些运算符可以替代标准的等号运算符，比如：

```
if "%varA%" equ "%varB%" (echo The values match!)
```

表3-3 使用比较运算符

运 算 符	描 述
<i>equ</i>	检查两个值是否相等，如果相等，则结果为真
<i>neq</i>	检查两个值是否不相等，如果不相等，则结果为真
<i>lss</i>	检查两个值之间的小于关系，如果值1小于值2，则结果为真
<i>leq</i>	检查两个值之间的小于等于关系，如果值1小于等于值2，则结果为真
<i>gtr</i>	检查两个值之间的大于关系，如果值1大于值2，则结果为真
<i>geq</i>	检查两个值之间的大于等于关系，如果值1大于等于值2，则结果为真

3.8 命令行迭代语句

如果需要重复地执行一条命令或一系列命令，可以使用 *for* 语句。*for* 语句具有非常强大的功能，如果你认为自己了解 *for* 语句而想跳过本节的内容，建议你放弃这种不明智的想法。本节讲述的 *for* 语句是专门为命令 *shell* 环境设计的，与你以前在其他程序设计语言环境中使用的 *for* 语句有很大的不同。命令行中的 *for* 语句主要用于在成组的文件与目录中进行迭代处理，并以行为基础分析文本文件、字符串以及命令的输出信息。

3.8.1 迭代的基础

命令 *shell* 有几种不同形式的 *for* 语句，最基本的 *for* 语句形式为：

```
for iterator do (statement)
```

这里，`iterator`用于控制`for`循环的执行。对`iterator`中每一个步骤或元素，都会执行特定的语句。它可以是单一的一条命令，也可以是使用命令管道、命令链、命令分组等技术组合起来的多条命令。

`iterator`通常包含一个初始化变量和一组需要反复执行的元素，比如需要遍历的一组文件或某范围内的一组值。初始化变量实质上是要使用的值的占位符，使用初始化变量时，应该注意如下几点。

- ❑ `iterator`变量只存在于`for`循环的上下文中。
- ❑ `iterator`变量名必须在`a~z`或者`A~Z`的范围内，比如`%%A`、`%%B`、`%%C`。
- ❑ `iterator`变量名是大小写敏感的，也就是说，`%%A`与`%%a`是不同的。

如表3-4中所示，用于`for`循环语句的不同结构具有特定的用途与形式。`for`语句初始化时，`iterator`变量，比如`%%B`，会被其真实值替代。这些值来自于`for`循环语句中指定的元素集，可以包含一组文件、一组目录、某范围内的一组值等。

表3-4 迭代的不同形式

迭代用途	语法格式
文件集合	<code>for %%variable in (fileSet) do statement</code>
目录集合	<code>for /D %%variable in (directorySet) do statement</code>
子目录中的文件	<code>for /R [path] %%variable in (fileSet) do statement</code>
遍历一系列的值	<code>for /L %%variable in (stepRange) do statement</code>
分析文本文件、字符串以及命令输出	<code>for /F [" options "] %%variable in (source) do statement</code>

真实场景 上表中提供的是`for`脚本中的不同迭代形式，也可以在命令行中交互式地使用`for`语句，这种情况下，应该使用`%variable`，而不是`%%variable`。除此之外，脚本中的`for`语句与命令行中使用的`for`语句在处理上是一致的。

3.8.2 遍历一系列值

使用`for`语句的传统方式是遍历某范围内的一系列值，并使用这些值执行相应任务。在命令shell中也可以做到这一点，其基本语法为：

```
for /L %%variable in (start,step,end) do (statement)
```

这种类型的`for`语句以如下方式运行。首先，命令shell对内部变量`start`、`step`、`end`进行初始化，将其赋值为应用中实际指定的值。之后，命令shell对`start`值与`end`值进行比较，如果`start`值可以按`step`中指定的值进行递增或递减操作，则条件为真（或按照另一种约定，条件为假），根据判别条件的真假来判断是否需要执行语句。在条件为真的情况下，命令shell使用`start`值执行语句，并根据指定的`step`值对初始值进行递增或递减操作，之后重复这一过程直至遍历所有的值，或者条件发生变化。在条件为假的情况下，命令shell退出`for`循环语句，并跳到脚本中的下一语句执行。

参考如下的实例，该实例对0到10进行计数，`step`值为2：

```
for /L %%B in (0, 2, 10) do echo %%B
```

该语句的输出为：

```
0
```

```
2
4
6
8
10
```

你也可以使用负的step值，使得for语句以值递减的方式执行。比如，在下面的实例中，对10到0进行计数，step值为-2：

```
for /1 %%B in (10, -2, 0) do echo %%B
```

该语句的输出为：

```
10
8
6
4
2
0
```

3.8.3 在成组的文件中迭代执行

在命令shell中，for语句更强大的功能是对文件与目录进行处理。处理成组的文件时，基本的for语句语法为：

```
for %%variable in (fileSet) do (statement)
```

这里，fileSet用于指定需要处理的文件集，文件集可以为如下的形式。

- 通过文件名指定的单独的文件，比如MyFile.txt。
- 通过文件名通配符指定的一组文件，比如*.txt。
- 通过多个文件名（使用空格分隔）指定的多个或多组文件，比如*.txt *.rtf *.doc。

了解这些基本规则后，使用for语句进行文件处理是很容易的。比如，如果想列出某应用程序目录中所有文本文件，可以在脚本中使用如下的命令：

```
for %%B in (C:\Working\*.txt) do (echo %%B)
```

这里，B为初始化变量，C:\Working*.txt指定了对C:\Working目录下所有文本文件进行处理。要循环执行的语句为echo %%B，通过该语句，在for循环的每次迭代中，命令shell都会展示%%B的当前值。上面语句的执行结果是，C:\Working目录中的文本文件列表被写入到输出中。

通过对上面的命令进行扩展，就可以列出所有.txt文件、.rtf文件与.doc文件，如下所示：

```
for %%B in (%AppDir%\*.txt %AppDir%\*.rtf %AppDir%\*.doc) do (echo %%B)
```

进一步地，还可以使用命令管道、命令链、命令分组等技术，如下所示：

```
for %%B in (%AppDir%\*.txt %AppDir%\*.rtf %AppDir%\*.doc) do (echo %%B &
move C:\Data)
```

在该命令中，列出了由AppDir变量指定的位置中所有.txt文件、.rtf文件与.doc文件，并将这些文件移动到C:\Data目录。

3.8.4 在目录中迭代执行

如果想操作目录，则可以使用如下的for语句格式：


```
for /d %%variable in (directorySet) do (statement)
```

这里，使用`directorySet`指定需要处理的目录组。对目录的迭代处理与对文件的迭代处理是一样的，所不同的是，此时指定的是目录路径，而非文件路径。如果想列出`%SystemRoot%`目录下的所有基目录，可以使用如下命令：

```
for /d %%B in (%SystemRoot%\*) do echo %%B
```

在Windows Server 2003中，所得到的部分结果列表类似于如下的形式：

```
C: \Windows\AppPatch
C: \Windows\Cluster
C: \Windows\Config
C: \Windows\Cursors
C: \Windows\Debug
```

注解 `for /d`循环对指定的目录集进行迭代处理，但不包括这些目录的子目录。要访问子目录（以及整个目录树结构），可以使用`for /r`循环，后面会对其进行讨论。

通过使用空格分隔目录名，可以指定多个基目录，如下所示：

```
for /d %%B in (%SystemRoot% %SystemRoot%\*) do echo %%B
```

上面的语句对`%SystemRoot%`目录及恰在该目录下的子目录进行检查。因此，上述命令执行结果列出的目录路径会以`C:\Windows`引导（前提是`C:\Windows`为系统目录）以及其他以前列出的目录。

文件迭代与目录迭代技术也可以结合起来，以便对目录集中的所有文件进行处理，如下所示：

```
for /d %%B in (%APPDATA% %APPDATA%\*) do (
@for %%C in ("%B\*.txt") do echo %%C)
```

上面的命令中，第一个`for`语句会返回`%APPDATA%`目录下的顶级目录列表，也包括`%APPDATA%`目录本身。第二个`for`语句则对这些目录中的所有文本文件进行迭代处理。要注意第二个`for`语句之前的`@`符号，该符号用于表明第二个`for`语句是嵌套的，同时也用于保证命令的正确执行。文件集的双引号（`" %%B*.txt "`）的作用是确保包含空格的目录与文件名被正确处理。

考虑到经常需要对目录及其子目录进行处理，命令shell提供了`for /r`语句。通过`for /r`语句，可以由`path`指定为起点的整个目录树进行处理，其语法为：

```
for /r [path] %%variable in (fileSet) do (statement)
```

上面的命令中，`path`设定了需要处理的目录树的起点，比如`C:\`。`path`并不是必需的，如果没有指定，则命令shell会将其假定为当前目录。

使用`for /r`语句（而不需要双重的`for`循环），可以对前面给出的实例进行扩展，以便列出C:盘中所有文本文件，如下所示：

```
for /r C:\ %%B in (*.txt) do echo %%B
```

可以看出，与双重的`for`循环相比，`for /r`语句简单而同时又更为强大。有时候甚至可以将`for /r`与`for /d`结合起来使用（而不需使用双重的`for`循环），如下面给出的实例，该实例列出了`%SystemRoot%`下的所有目录及其子目录：

```
for /r %SystemRoot% /d %%B in (*) do echo %%B
```

3.8.5 分析文件的内容与输出

除了通过指定文件名与目录名进行处理之外,你也可以对文件内容与命令输出信息进行处理。要做到这一点,可以使用如下的语法格式:

```
for /f ["options"] %%variable in (source) do (statement)
```

上面的命令语法中, *options* 用于设置文本匹配选项, *source* 指定了文本的来源(可以是文本文件、字符串或命令的输出信息), *statement* 指定了在文本匹配时要执行的命令。*source* 中的每行文本被命令 *shell* 当作一个记录处理,不同的字段由特定的字符分隔开(比如制表符或空格,默认情况下是空格)。通过替换技术,命令 *shell* 在执行时会使用变量的实际值来替代占位符变量。

参考某源文件中如下一行文本:

```
William Stanek Engineering Williams@adatum.com 3408
```

如果将该行文本看成一个记录,可以划分为如下5个字段。

- 首名: William。
- 尾名: Stanek。
- 系别: Engineer。
- 电子邮件地址: Williams@adatum.com。
- 电话扩展: 3408。

要分析该行文本以及相关文件中的其他类似文本行,可以使用类似如下的 *for* 语句:

```
for /f "tokens=1-5" %%A in (current-users.txt) do (
@echo Name: %%A %%B Depart: %%C E-mail : %%D Ext: %%E)
```

该命令指定了要处理的前5个字段(*token*字段,默认情况下由空格或制表符分隔开),并由 *iterator* 变量标识, *iterator* 变量以 *%%A* 开始。也就是说,第一个字段是 *%%A*,第二个字段是 *%%B*,依此类推。上面命令的输出类似于如下的形式:

```
Name: William Stanek Depart: Engineering E-Mail: Williams@adatum.com Ext:
3408
```

表3-5展示了可用选项(用于文件内容与命令输出信息分析)的完整列表,包括实例及其描述。

表3-5 用于文件内容与命令输出信息分析的选项

选 项	选项描述	实 例	实例描述
<i>eol</i>	设置行尾注释字符,行尾注释字符后的所有数据都被命令 <i>shell</i> 看成注释	<i>eol=#</i>	将#设置为行尾注释字符
<i>skip</i>	设置文件起始处跳过的行数	<i>skip=5</i>	通知命令 <i>shell</i> 跳过源文件中的1到5行
<i>delims</i>	设置各字段之间的分隔符,默认情况下为制表符或空格	<i>delims=,,:;</i>	指定逗号、句点、分号为分隔符
<i>tokens</i>	为每一源行设置令牌字段,如果以 <i>a</i> 或 <i>A</i> 作为起始的迭代变量,则至多可以指定26个令牌。默认情况下,只对第一个令牌进行检查	<i>tokens=1,3</i> <i>tokens=2-5</i>	第一个实例将令牌字段设置为使用1与3,第二个实例将令牌字段设置为使用2、3、4、5
<i>usebackq</i>	规定可以在源指定符中使用引号:对文件名使用双引号,对命令使用反引号,对字符串使用单引号	<i>usebackq</i>	激活该选项

要了解这些选项如何使用，可以参考下面的实例：

```
for /f "skip=3 eol=; tokens=3-5" %%C in (current-users.txt) do (
@echo Depart: %%C E-mail : %%D Ext: %%E)
```

上面的命令中，使用了3个选项。其中，*skip*选项用于跳过文件的前3行，*eol*选项用于将行尾注释字符设置为分号 (;)，*tokens*选项则规定，token3到token5应该放置到iterator变量中，且以%%C开始。通过使用tokens，可以以很多种方式来指定要使用的字段，下面给出了一些实例。

- **tokens=2,3,7**。使用字段2、3、7。
- **tokens=3-5**。使用字段3、4、5。
- **tokens=***。把每一行当作整体进行处理，而不分割为字段。

在处理文本文件时，需要注意的是，文本文件中所有空行都会被忽略。此外，可以通过通配符或空格分隔的文件名列列表来指定多个源文件，比如：

```
for /f "skip=3 eol=; tokens=3-5" %%C in (data1.txt data2.txt) do (
@echo Depart: %%C E-mail: %%D Ext: %%E)
```

如果文件名包含空格，或者想执行某条命令，则可以使用*usebackq*选项与引号，比如：

```
for /f "tokens=3-5 usebackq" %%C in ("user data.txt") do (
@echo Depart: %%C E-mail: %%D Ext: %%E)
```

或者：

```
for /f "tokens=3-5 usebackq" %%C in ('type "user data.txt"') do (
@echo Depart: %%C E-mail: %%D Ext: %%E)
```

提示 要记住的是：反引号 (`) 可用于封装命令，而单引号 (') 则用于封装字符串。显然，两个符号在外观上是非常类似的。在标准键盘上，反引号 (`) 与波浪线 (~) 在同一个键上，单引号 (') 则与双引号 (") 在同一个键上。

注解 上面第二个实例就是一个使用反引号封装命令的实例，其中使用TYPE命令将文件内容写入到标准输出。

此处谈及引号时，主要是将其用于字符串与变量名的处理。比如，可以使用双引号将字符串或变量名封装起来，以确保命令shell对其进行正确处理，而不需要再使用*usebackq*选项。

参考如下的实例：

```
set value=All,Some,None
for /f "delims=, tokens=1,3" %%A in ("%VALUE%") do (echo %%A %%B)
```

其输出为：

```
All None
```

3.9 创建子程序与过程

通常，命令shell逐行执行脚本，从文件的起始行开始执行，执行到文件末尾结束。在实际的应用中，也可以改变这种执行顺序，这是通过如下的技术实现的。

- **子程序**。通过使用子程序，可以使得命令shell跳转到当前脚本内的某个标记处，并从该处执行

直至文件结束。

- **过程**。通过使用过程，可以调用其他脚本，并在调用的脚本执行完毕后将控制权返回到原脚本中调用其他脚本的语句的下一行。

可以看出，子程序与过程的主要区别在于所要完成的任务。此外，传递给脚本的参数可以直接提供给`goto`子程序使用，而被调用过程内的参数列表则被改变为以过程名（而不是脚本名）作为参数0（%0）。

3.9.1 使用子程序

子程序包含下面两个部分。

- **goto语句**。指定了命令shell将要跳转到的子程序。
- **标号**。指定了子程序的开始。

参考如下子程序调用的实例：

```
if "%1"=="1" goto SUB1
```

在上面的命令中，如果第一个参数为1，则调用名为`SUB1`的子程序，同时命令shell会跳转到相应的子程序标号处。要创建一个标号，可以在某行命令前输入一个关键字，并以分号引导，比如：

```
:SUB1
```

尽管标号可以包含任意有效类型的字符，但通常都会使用字母数字型的字符。这样，在自己或其他人查阅代码时会更容易阅读和理解。

使用`goto`语句时，命令shell会在目标标号的下一行处开始执行脚本，直至文件结束，除非执行过程中遇到其他过程调用或`goto`语句。如果标号在脚本当前位置之前，则命令shell可以回跳到脚本中更靠前的部分，这可能导致死循环的出现（除非有某种控制措施来绕过`goto`语句），下面给出一个死循环的实例：

```
:START
```

```
.
```

```
.
```

```
.
```

```
goto START
```

如果标号在`goto`语句之后，则命令shell可以绕过一些命令直接跳到脚本中的某一部分，比如：

```
goto MIDDLE
```

```
.
```

```
.
```

```
.
```

```
:MIDDLE
```

这一实例中，命令shell直接跳转到`:MIDDLE`标号处继续执行脚本，直至文件结束，而不能回去执行在`goto`语句与`:MIDDLE`标号之间的那些命令（除非使用了其他`goto`语句）。

有些时候，可能不需要执行脚本的剩余部分，而是在执行子程序后直接退出脚本。要做到这一点，可以在脚本中创建一个退出标号，并在子程序结束后跳转到该标号处，比如：

```
goto MIDDLE
```

```
.
```

```
.
```

```

.
:MIDDLE
.
.
goto EXIT
.
.
:EXIT

```

命令清单3-4展示了一个使用`goto`语句与标号的详尽实例。在该实例中，脚本的第一个参数值决定了执行哪个子程序。第一个`if`语句用于在没有参数传递给脚本时显示一条错误消息并退出，其后的`goto EXIT`语句则用于处理无效参数传递给脚本的情况（该实例中只是跳转到`:EXIT`标号处）。

命令清单3-4 使用`goto`语句

```

>@echo off
if "%1"==" " (echo Error: No parameter passed with script!) & (goto
EXIT)
if "%1"=="1" goto SUBROUTINE1
if "%1"=="2" goto SUBROUTINE2
if "%1"=="3" goto SUBROUTINE3
goto EXIT

:SUBROUTINE1
echo In subroutine 1
goto EXIT

:SUBROUTINE2
echo In subroutine 2
goto EXIT

:SUBROUTINE3
echo In subroutine 3
goto EXIT

:EXIT
echo Exiting...

```

提示 如果调用了一个不存在的标号，则脚本文件搜索到最后会给出一条错误信息，之后脚本退出，而不再执行后续跟随的命令。熟练使用子程序功能的命令shell程序员，比如我，习惯于使用`goto EXIT`语句并提供一个实际的`:EXIT`标号（参考前面的实例）。但实际上命令解释器也支持一个`:EOF`标号，该标号将命令shell的控制权转移到文件末尾，并充当脚本退出的标志（而不需要单独定义一个标号）。

3.9.2 使用过程

通过使用过程，可以在脚本中调用其他脚本，而又不会退出原脚本。进行过程调用时，命令shell会执行调用的脚本，逐一执行其中的命令，结束后返回到原脚本，并从过程调用语句的下一行语句开

始执行。参考如下的实例：

```
if "%1"=="1" call system-checks
if "%1"=="2" call C:\scripts\log-checks
```

警告 如果没有使用call语句，但又在原脚本中引用了其他脚本名，则命令shell会执行引用的脚本，但执行完毕后控制权不会返回到调用者。

上面的命令中，第一个调用针对的脚本应该处在当前目录或命令路径上。第二个调用针对的脚本文件路径为c:\scripts\log-checks。

在使用过程调用时，传递给原脚本的参数在传递给调用的脚本时会会有所变化：参数列表更新为包含过程名，并将其作为参数0（%0）。那些为过程所专用的参数则在调用的脚本文件执行结束之前一直有效，直到控制权返回到原脚本。

也可以向调用的脚本传递参数，比如：

```
set Arg1=mailer1
set Arg2=dc2
set Arg3=web3
call system-checks Arg1 Arg2 Arg3
```

执行上述命令后，变量`Arg1`、`Arg2`、`Arg3`在调用的脚本中都是可用的。



Part 2

第二部分

使用命令行管理 Windows 系统

本 部 分 内 容

- 第 4 章 部署 Windows 服务器
- 第 5 章 管理 Windows 系统
- 第 6 章 事件记录、追踪与监控
- 第 7 章 进程监控与性能维护
- 第 8 章 管理事件与性能日志
- 第 9 章 计划任务的自动运行

与Windows Vista相比，使用Windows Server 2008时会有更多的配置选项。本章将详细讲述这些选项，目标是帮助读者使用命令行工具（或命令行工具与图形界面工具的组合）来部署Windows服务器。

4.1 服务器配置管理

使用Windows Server 2008时，在部署新的服务器之前，应该认真规划服务器的体系结构。根据不同服务器的具体需求，对软件配置、硬件配置进行适当的修改。Windows Server 2008的安装可以采用如下两种类型。

- **Full-server安装。**这种安装类型提供了全部的功能。这种类型的安装中，可以使用角色、角色服务以及附加软件功能的任意有效组合来配置服务器，并提供了一个全面的用户界面来对服务器进行管理。在部署Windows Server 2008时，如果服务器角色会随时间和实际需求变化，建议选择这种安装模式。
- **Core-server安装。**这种安装类型提供了最基本的功能。这种类型的安装中，只可以使用有限的角色，提供的用户界面的功能也是最基本的（用于实现对服务器的本地管理）。如果需要使用专用的服务器来充当特定的服务器角色（或几种角色的组合），也希望降低其他服务带来的负载，建议选择这种安装模式。

在安装Windows Server 2008时，可以根据需要选择安装类型。要注意的是，尽管你可以使用任何允许使用的远程管理技术来对这两种安装类型进行远程管理，但在本地控制台进行管理时，这两种安装类型是完全不同的。因此，要记住如下一些区别。

- 进行full-server安装时，有一个完整的用户界面，其中包含了完整的桌面环境（用于从本地控制台对服务器进行管理）。同时，可以使用允许使用的角色、角色服务与功能的任意组合来部署服务器。
- 进行core-server安装时，仅支持有限的一些角色与角色组合方式。支持的角色包括：活动目录服务（AD DS）、域名服务（DNS）Server、动态主机配置协议（DHCP）Server、文件服务以及打印服务。此外，在当前的实现中，这种安装模式不提供对服务器应用程序的支持。

提示 要安装服务器，可以使用Windows Setup（setup.exe）。安装过程中，在“您想将Windows安装在何处？”页面，你可以通过按Shift+F10键来访问命令提示符。通过这种方法，你可以访问Windows Server 2008标准安装过程中可用的很多类似的命令行工具，包括DiskPart。

由于core-server安装只提供最基本的用户界面与有限的桌面环境，其管理方式与full-server安装有较大的差别。最基本的用户界面包括如下几个。

- Windows登录屏幕，用于用户的登录与注销。
- 命令提示符，用于通过命令行进行管理。
- 记事本，用于编辑文件。
- Regedit命令，用于管理注册表。
- 任务管理器，用于管理任务与启动新任务。

启动一台core-server安装的服务器时，可以使用Windows登录屏幕来登录系统（与full-server安装时一样）。在域中，对登录服务器有一些标准的约束机制，具有适当的用户权限与登录许可权限的用户才可以登录服务器。在未充当域控制器的、工作组环境中的服务器上，可以使用NET USER命令来添加用户，使用NET LOCAL GROUP命令来将用户添加到本地组，以便于本地登录。

登录到core-server模式安装的服务器后，可以操作一个受限的桌面环境与管理权限的命令提示符。你可以使用该命令提示符来对服务器进行管理，如果无意间关闭了该提示符，则可以通过如下步骤来启动一个新的命令提示符。

- (1) 按Ctrl+Shift+Esc键，弹出“任务管理器”。
- (2) 在“应用程序”选项卡中单击“新任务”。
- (3) 在“创建新任务”对话框中，在“打开”行，键入cmd，之后单击“确定”。

通过与上面类似的步骤，也可以打开其他命令提示符窗口。尽管你可以在上面的步骤(3)中键入notepad.exe或regedit.exe（而非cmd）来分别打开记事本与注册表，但也可以在命令提示符中输入notepad.exe或regedit.exe来分别打开记事本与注册表。

登录服务器后，可以通过按Ctrl+Alt+Delete键随时显示Windows登录屏幕，core-server安装模式的Windows登录屏幕与full-server安装模式的Windows登录屏幕具有同样的选项，包括锁定计算机、切换用户、注销登录、修改密码、启动任务管理等。在命令提示符中，可以发现所有用于服务器管理的标准命令与命令行工具。然而，要记住的是，命令、命令行工具与程序的正确运行有一个前提——它们所依赖的组件必须已经安装在系统中。

尽管core-server安装模式仅支持有限的角色与角色服务，但可以安装大多数功能。然而，例外的情况是，你无法安装那些依赖于微软.NET框架的功能。因为原始的core-server实现不支持.NET框架，因而不能添加Windows PowerShell等功能。随着升级包与服务包的安装，这一限制因素也可能会改变。此外，与任意的full-server安装一样，你也可以通过终端服务远程管理core-server安装方式的服务器。

4.2 使用角色、角色服务与功能

安装了服务器之后，可以通过安装并配置如下的组件来对服务器配置进行管理。

- **服务器角色。**服务器角色是相关的软件组件集，主要作用是使得服务器可以为用户及网络上的其他计算机提供特定的功能。Windows服务器可以单独充当一种角色，比如文件服务，也可以同时充当多种角色。
- **角色服务。**为服务器角色提供功能支持的软件组件。有一些服务器角色只有单一的功能，安装了该服务器角色也就意味着安装了该功能。大多数服务器角色都有多重的、相关的角色服务，安装时可根据需要选择安装哪一种角色服务。

- 功能。提供附加功能的软件组件，功能的安装与删除独立于角色与角色服务。计算机可以安装多重的功能，也可以完全不安装，这依赖于具体的配置与应用需求。

要对系统中的角色、角色服务、功能等进行管理，可以使用ServerManagerCmd（命令行形式的管理工具）或服务器管理器（图形界面的管理工具）。由于在同一时间只能使用ServerManagerCmd与服务器管理器中的一个来添加或删除组件，因此，在使用服务器管理器的添加或删除功能时，就无法再使用ServerManagerCmd。

表4-1中列出了可以在Windows Server 2008上部署的主要角色与相关的角色服务。除Windows Server 2008默认包含的角色与功能之外，ServerManagerCmd与服务器管理器还支持对来自微软下载中心的附加的角色与功能的整合，这些附加的功能与角色是以Windows Server 2008更新包形式提供的（可选的）。

表4-1 Windows Server 2008中的主要角色与相关的角色服务

角 色	描 述
Active Directory证书服务 (AD CS)	AD CS提供了必要的功能，用于发布与撤销用户、客户端计算机、服务器的数字证书。主要包含如下角色服务：认证中心、认证中心Web注册、在线证书状态协议、微软简单证书注册协议（MSCEP）
Active Directory域服务 (AD DS)	AD DS提供了必要的功能，用于存储用户、组、计算机以及网络上其他客体对象的相关信息，并使得这些信息对用户与计算机是可用的。域控制器使得网络用户与计算机可以对网络上允许访问的资源进行访问
Active Directory联合身份验证服务 (AD FS)	AD FS扩展和补充了AD DS的认证与访问管理功能（从域扩展到万维网），主要包含如下一些角色服务与子服务：联合身份验证服务、联合身份验证服务代理、AD FS Web代理、Claims-aware代理、基于Windows 令牌的代理
Active Directory轻型目录服务 (AD LDS)	AD LDS为那些激活了目录功能、但不需要AD DS也不需要部署在域控制器内的应用程序提供了数据存储功能，AD LDS不包括附加的角色服务
Active Directory权限管理服务 (AD RMS)	AD RMS为受保护的电子邮件消息、文档、内联网Web页面以及其他类型的相关文件提供了可控的访问功能，主要包含如下一些角色服务：Active Directory权限管理服务、实体联合身份验证支持
应用程序服务器	应用程序服务器使得服务器可以充当分布式应用程序（使用ASP.NET、Enterprise Services以及.NET Framework 3.0等技术构建）的宿主。包含了多种角色服务，在 <i>Internet Information Server 7.0 Administrator's Pocket Consultant</i> （Microsoft Press, 2007）一书中对其进行了详细讨论
动态主机配置协议 (DHCP) 服务器	DHCP提供了对IP地址的集中控制功能，DHCP服务器可以为网络上的其他计算机分配动态IP地址，也可以进行静态的TCP/IP设置。不包含附加的角色服务
DNS服务器	DNS是一种域名解析系统，可以将计算机名解析为IP地址，在活动目录域中，DNS是进行域名解析的基础。不包含附加的角色服务
传真服务器	传真服务器为企业中传真的收发提供了集中化的控制。传真服务器可以充当传真业务的网关，以便对传真资源（比如传真作业、传真报告，以及部署在网络或服务器上的传真设备）进行管理。不包含附加的角色服务
文件服务	文件服务用于对文件及其在网络上可用与复制的方式进行管理，很多服务器角色都需要某种类型的文件服务。包含如下一些角色服务与子服务：文件服务器、分布式文件系统、DFS命名空间、DFS复制、文件服务器资源管理器、网络文件系统（NFS）服务、Windows 搜索服务、Windows Server 2003文件服务、文件复制服务（FRS）、索引服务
网络策略和访问服务 (NPAS)	NPAS为路由及网络远程访问的管理提供了基本服务。包含如下一些角色服务：网络策略服务器（NPS）、路由与远程访问服务（RRAS）、远程访问服务、路由、健康注册授权、主机信任状授权协议（HCAP）等

(续)

角 色	描 述
打印服务	打印服务用于对网络打印机与打印驱动程序进行管理。包含如下一些角色服务：打印服务器、LPD服务、Internet打印
终端服务	终端服务使得用户可以运行安装在远程计算机上的Windows应用程序。用户运行终端服务器上的应用程序时，程序的执行与处理是在该服务器上进行的，只有来自应用程序的数据是在网络上传输的。包含如下一些角色服务：终端服务器、TS授权、TS Session Broker、TS网关、TS Web Access
通用描述、发现和集成 (UDDI) 服务	UDDI为组织内部或组织之间共享Web服务信息提供了便利。包含如下一些角色服务：UDDI服务数据库、UDDI服务Web应用程序
Web服务器 (IIS)	Web服务器 (IIS) 为网站与基于Web的应用程序提供了宿主平台。以Web服务器为宿主机的网站可以包含静态内容与动态内容，部署在Web服务器上的应用程序可以使用ASP.NET与.NET Framework 3.0进行构建。部署Web服务器时，可以使用IIS 7模块与管理工具对服务器配置进行管理。包含了多种角色服务，在 <i>Internet Information Server 7.0 Administrator's Pocket Consultant</i> 一书中对其进行了详细讨论
Windows部署服务	Windows部署服务用于在企业中部署Windows主机。包含如下的角色服务：部署服务器、传输服务器
Windows SharePoint服务	通过信息的交互，Windows SharePoint服务为团队协作提供了便利。SharePoint服务器实际上是一台运行了完整IIS安装的Web服务器，其中使用了管理型应用程序，提供了必要的协作功能支持
Windows Server更新服务	通过Windows Server更新服务，可以使用集中式服务器来为组织内的不同计算机分发更新包，而不再需要对每台主机进行单独更新

表4-2列出了可以在Windows Server 2008上部署的主要功能。与以前的Windows发行版不同，在Windows Server 2008上，有些重要的服务器功能不是自动安装的。比如，要使用操作系统内置的备份与恢复功能，就必须添加Windows Server Backup功能。

表4-2 针对Windows Server 2008的主要功能

功 能	描 述
.NET Framework 3.0	为应用程序开发提供了.NET Framework 3.0 API。附加的子功能包括：.NET框架3.0功能、XPS查看器、Windows Communication Foundation (WCF) 激活组件
Bitlocker驱动器加密	提供了基于硬件的安全功能，通过全盘加密对数据进行保护，防止操作系统离线时的磁盘信息泄露。安装了可信平台模块 (TPM) 的计算机可以在Startup Key或TPM-only两种模式中使用Bitlocker驱动器加密，这两种模式都提供了早期的完整性验证功能
后台智能传送服务 (BITS) 服务器扩展	提供了后台智能传送服务。该功能激活时，Windows服务器充当BITS服务器，可以接收客户端上传的文件，但该功能对使用BITS下载的客户是不必要的
连接管理器管理工具包 (CMAK)	提供了用于生成连接管理器配置文件的功能
桌面体验	在服务器上提供了附加的Windows Vista桌面功能。Windows Vista增加的功能包括：Windows Media Player、桌面主题、Windows相册管理。尽管这些功能使得用户可以向使用桌面计算机一样便利地使用服务器，但同时也降低了服务器的总体性能
故障转移集群	提供了集群功能，使得多台服务器可以一起工作，从而为服务与应用程序提供了高可用性。很多类型的服务都可以进行集群处理，包括文件与打印服务等。此外，消息与数据库服务器也是可以集群处理的理想对象

(续)

功 能	描 述
组策略管理	安装组策略管理控制台 (GPMC)，提供了对组策略的集中化管理功能
Internet打印客户端	提供了相应的功能，使得客户端可以使用HTTP连接到Web打印服务器上的打印机
Internet存储名称服务器 (iSNS)	为Internet SCSI (iSCSI) 设备提供了管理与服务器功能，使得服务器可以处理来自iSCSI设备的注册请求、注册取消请求以及查询请求
行式打印机远程 (LPR) 端口监控器	安装LPR端口监控器，使得系统可以向连接到基于Unix的计算机的设备进行打印
消息队列	为分布式消息队列提供了管理与服务器功能，同时提供了一组相关的子功能
多路径I/O (MPIO)	为使用多数据路径将数据保存到存储设备提供了必要的功能
网络负载均衡 (NLB)	通过将入站的应用程序请求在一组参与的服务器中进行分布，NLB为基于IP的应用程序与服务提供了故障转移支持与负载均衡功能。Web服务器是进行负载均衡调试的理想对象
对等名称解析协议 (PNRP)	提供了链接-本机多点传送名称解析 (LLMNR) 功能，该功能运行进行端对端的名称解析服务。安装这一功能后，运行在服务器上的应用程序可以使用LLMNR进行注册与名称解析
远程协助	允许远程用户连接到服务器，并提供或接受远程协助
远程服务器管理工具 (RSAT)	安装角色管理工具与功能管理工具，这两款工具可用于对其他Windows Server 2008系统进行远程管理。每款工具的安装都有一些可用的选项，你也可以按照顶层范畴或子范畴来安装工具
可移除存储管理器 (RSM)	安装可移动存储管理器工具，该工具可用于管理可移动的介质与可移动的介质设备
RPC Over HTTP 代理	安装代理，用于将RPC消息从客户端应用程序转发到服务端（基于HTTP协议）。对使用VPN连接访问服务器的客户端而言，基于HTTP的RPC提供了一种可替代的方案
简单TCP/IP 服务	安装附加的TCP/IP服务，包括Character Generator、Daytime、Discard、Echo以及Quote of the Day等
简单邮件传输协议 (SMTP) 服务器	SMTP是一种用于控制电子邮件消息的传输与路由的网络协议。激活这一功能时，Windows Server 2008可以充当基本的SMTP服务器，不过如果需要功能完备的SMTP服务器解决方案，则需要安装Microsoft Exchange Server 2007等消息服务器
简单网络管理协议 (SNMP) 服务	SNMP是一种用于简化TCP/IP网络管理的协议，如果所在网络安装了与SNMP兼容的设备，则可以使用SNMP进行集中化的网络管理也可以通过网络管理软件并借助SNMP协议进行网络监控
SANs存储管理器	安装SANs存储管理器控制台。该控制台为存储区域网络提供了集中化的管理接口，通过该工具，可以查看存储子系统、创建并管理逻辑单元数，也可以用于管理iSCSI目标设备。SAN设备必须支持可视化磁盘服务
基于UNIX应用程序的子系统 (SUA)	为Windows系统提供了运行基于Unix的应用程序的功能。可以从微软下载网站下载附加的管理工具
Windows Internal Database	安装SQL Server 2005精简版，使得服务器可以使用关系数据库，以便那些需要内部数据库的Windows角色与功能的正确运作，比如AD RMS、UDDI服务、Windows Server更新服务 (WSUS)、Windows SharePoint服务、以及Windows系统资源管理等

(续)

功 能	描 述
Windows Powershell	安装Windows Powershell, 为Windows系统管理提供了增强的命令行环境
Windows进程激活服务	为那些分布式的Web应用程序(使用HTTP以及其他协议)提供支持
Windows Server Backup	为操作系统、系统状态以及存储在服务器上的其他数据提供备份与恢复功能
Windows系统资源管理者(WSRM)	用于对资源的使用进行管理(以每个处理器为基础)
WINS服务器	WINS是一种名解析服务, 可以将计算机名解析为IP地址。安装这一功能使得计算机可以充当WINS服务器
无线网络连接	该功能使得服务器可以使用无线网络连接与配置文件

在设计服务器管理器与ServerManagerCmd时, 可扩展性是微软努力达到的目标。这样设计的好处在于, 为操作系统提供补充的角色、角色服务以及功能是比较容易的。在微软的下载网站, 提供了一些附加的组件下载, 包括Windows媒体服务(用于Windows Server 2008)以及Windows SharePoint Services 3.0。

通过如下步骤, 可以下载这些组件, 并对其进行安装与配置。

- (1) 从微软下载网站下载安装程序包。典型情况下, 这些安装程序包是以一组微软更新独立安装包(.msu)文件的形式提供的。
- (2) 双击每个安装程序包, 对其进行注册以便使用。
- (3) 如果服务器上正在运行服务器管理器, 对其进行刷新或重启, 使得新组件可用。
- (4) 在服务器管理器中, 使用适当的向导来安装并配置补充的角色、角色服务与功能。

4.3 管理角色、角色服务与功能

如果需要从命令行对服务器配置进行管理, ServerManagerCmd是可以使用的主要工具。通过该工具, 不仅可以实现对角色、角色服务与功能的添加或删除, 还可以使用该工具查看这些软件组件的详细配置与状态等相关信息。

4.3.1 ServerManagerCmd 基础

使用ServerManagerCmd时, 你应该使用一个增强的管理员命令提示符。在该提示符中, 可以使用如下的参数及命令行语法格式对角色、角色服务与功能等进行管理。

- ❑ **Servermanagercmd -query**。列出服务器中角色、角色服务与功能的当前状态。如果指定了SaveFile.xml文件, 则查询结果会以XML格式显示并保存到该文件中。此外, 也可以使用-q参数来替代-query参数。

```
ServerManagerCmd -query [SaveFile.xml] [-logPath LogFile.txt]
```

- ❑ **Servermanagercmd -install**。安装指定的角色、角色服务与功能。通过-AllSubFeatures或-A参数, 可以安装指定组件的所有从属的角色服务与功能。-Setting参数或-S参数则用于将某些配置选项设置为特定的值。此外, 也可以使用-i参数来替代-install参数。

```
ServerManagerCmd -install ComponentName
[-setting SettingName=SettingValue] [-allSubFeatures]
[-resultPath Results.xml] [-restart] | -whatIf]
[-logPath LogFile.txt]
```

- **Servermanagercmd -inputPath**。添加或删除在XML answer文件中指定的角色、角色服务与功能，也可以使用-ip参数来替代-inputPath参数。

```
ServerManagerCmd -inputPath AnswerFile.xml [-resultPath
[Results.xml]] [-restart] | -whatIf]
[-logPath LogFile.txt]
```

- **Servermanagercmd -remove**。删除指定的角色、角色服务与功能，也可以使用-r参数来替代-remove参数。

```
ServerManagerCmd -remove ComponentName [-resultPath Results.xml]
[-restart] | -whatIf] [-logPath LogFile.txt]
```

- **Servermanagercmd -version**。列出当前使用的ServerManagerCmd的版本信息，也可以使用-v参数来替代-version参数。

```
ServerManagerCmd -version
```

每一个主要的参数都可以接受附加的参数与参数值，如下所示。

- 使用-LogPath参数或-L参数将错误信息记录到指定的日志文件。
- 使用-Restart参数自动重启计算机（适用于完成某操作时需要重启的情况）。
- 使用-ResultPath参数或-Rp参数将标准输出结果写入到XML格式的文件中。
- 使用-Whatif参数或-W参数显示命令执行时的操作。

可以使用的参数值包括如下几个。

- **AnswerFile.xml**。使用XML格式的answer文件来确定要添加或删除的组件。
- **ComponentName**。指定要管理的角色、角色服务与功能。
- **LogFile**。设置文本文件名（将错误信息写入到其中）。
- **Results.xml**。将安装或删除操作的结果保存到指定的XML格式的文件中（不影响结果的正常显示）。
- **SaveFile.xml**。将标准输出结果保存到指定的XML格式的文件中（不影响结果的正常显示）。需要注意的是，这些结果中不包括错误信息，错误信息被单独写入到标准错误输出中。
- **SettingName**。通过名字识别必需的设置。
- **SettingValue**。为某一元素设置具体的配置值。

大多数可以安装的角色、角色服务与功能都有相应的组件名。通过组件名可以识别具体的组件，以便于在命令行中对其进行操作，对于从微软网站下载并安装的补充性的组件同样如此。

表4-3中层次化地列出了组件名及其相关的角色、角色服务与子组件。安装某角色时，通过使用-AllSubFeatures参数，可以安装该角色下列出的所有从属的角色服务与功能。安装某角色服务时，通过使用-AllSubFeatures参数，可以安装该角色服务下列出的所有从属的功能。

表4-3 关键角色与角色服务的组件名

组 件 名	角 色	服 务	功 能
AD-Certificate ADCS-Cert-Authority ADCS-Web-Enrollment ADCS-Online-Cert ADCS-Device-Enrollment	活动目录证书服务	证书授权 证书授权Web注册 在线响应者 网络设备注册服务	
活动目录域服务			
ADCS-Domain-Controller ADDS-Identity-Mgmt ADDS-NIS ADDS-NIS ADDS-IDMU-Tools		活动目录域控制器 UNIX实体管理	网络信息服务器 口令同步 管理工具
活动目录联盟服务			
ADFS-Federation ADFS-Proxy ADFS-Web-Agents ADLDS DHCP DNS Fax	Active Directory 轻型目录服务 DHCP服务器 DNS服务器 Fax服务器	联合身份验证服务 联合身份验证服务代理 AD FS Web代理	
文件服务			
FS-FileServer FS-DFS FS-DFS-Namespace FS-DFS-Replication FS-Resource-Manager FS-NFS-Services FS-Search-Services FS-Win2003-Services FS-Replication FS-Indexing-Services Hyper-V NPAS NPAS-Policy-Server NPAS-RRAS-Services NPAS-RRAS NPAS-Routing NPAS-Health	Hyper-V 网络策略与访问服务	文件服务器 分布式文件系统 文件服务器资源管理器 网络文件系统服务 Windows搜索服务 Windows Server 2003文件服务 网络策略服务器 路由与远程访问服务 健康注册授权	DFS命名空间 DFS复制 文件复制服务 索引服务 远程访问服务 路由

(续)

组 件 名	角 色	服 务	功 能
NPAS-Host-Cred	打印服务	主机凭据授权协议	
Print-Services			
Print-Server		打印服务器	
Print-LPD-Service		LPD服务	
Print-Internet		Internet打印	
Terminal-Services	终端服务		
TS-Terminal-Server		终端服务器	
TS-Licensing		TS授权	
TS-Session-Broker		TS Session Broker	
TS-Gateway		TS网关	
TS-Web-Access	Windows部署服务	TS Web Access	
WDS			
WDS-Deployment		部署服务器	
WDS-Transport		传输服务器	

表4-4层次化地列出了一些功能、子功能及其对应的组件名。安装某功能时，可以使用-AllSubFeatures参数来安装该功能下所有从属的二级子功能与三级子功能。安装二级功能时，可以使用-AllSubFeatures参数来安装该二级功能下所有从属的三级子功能。

注解 功能后跟随的星号(*)表达的含义是：该功能没有列出从属的子功能，而通常情况下，这些子功能是通过-AllSubFeatures参数一起安装的。

表4-4 关键功能与子功能的组件名

组 件	功 能	二级子功能	三级子功能
NET-Framework*	.NET Framework 3.0功能		
BitLocker	BitLocker驱动器加密		
BITS	BITS服务器扩展		
CMAK	连接管理器管理工具		
Desktop-Experience	桌面体验		
Failover-Clustering	Failover集群		
GPMC	组策略管理控制台		
Internet-Print-Client	Internet打印客户端		
ISNS	Internet存储名服务器		
LPR-Port-Monitor	LPR端口监控器		
MSMQ*	消息队列		
Multipath-IO	多路径I/O		
NLB	网络负载均衡		
PNRP	对等名称解析协议		
qWave	Windows音频/视频体验质量		

(续)

组 件	功 能	二级子功能	三级子功能
Remote-Assistance	远程维护	角色管理工具	
RDC	远程差分压缩		
RSAT	远程服务器管理工具		
RSAT-Role-Tools			
RSAT-ADCS*			活动目录证书服务工具
RSAT-ADDS*			活动目录域服务工具
RSAT-ADLDS			活动目录轻量级目录服务工具
RSAT-RMS			活动目录权限管理服务工具
RSAT-DHCP			DHCP服务器工具
RSAT-DNS-Server			DNS服务器工具
RSAT-Fax			传真服务器工具
RSAT-File-Services*			文件服务工具
RSAT-NPAS*			网络策略与访问服务工具
RSAT-Print-Services			打印服务工具
RSAT-TS*			终端服务工具
RSAT-UDDI			UDDI服务工具
RSAT-Web-Server			Web服务器 (IIS) 工具
RSAT-WDS			Windows部署服务工具
RSAT-Feature-Tools		功能管理工具	
RSAT-BitLocker			BitLocker驱动器加密工具
RSAT-Bits-Server			BITS服务器加密工具
RSAT-Clustering			Failover集群工具
RSAT-NLB			网络负载均衡工具
RSAT-SMTP			SMTP服务器工具
RSAT-WINS			WINS服务器工具
Removable-Storage	可移除存储管理器		
RPC-over-HTTP-Proxy	运行于HTTP代理之上的RPC		
Simple-TCP/IP	简单TCP/IP服务		
SMTP-Server	SMTP服务器	SNMP服务 SNMP WMI提供者	
SNMP-Services	SNMP服务		
SNMP-Service			
SNMP-WMI-Provider			
Storage-Mgr-SANS	用于SAN的存储管理器		
Substem-UNIX-Apps	用于基于UNIX应用程序的子系统		
Telnet-Client	Telnet客户端		
Telnet-Server	Telnet服务器		
TFTP-Client	TFTP客户端		

(续)

组 件	功 能	二级子功能	三级子功能
Windows-Internal-DB	Windows Internal Database		
PowerShell	Windows PowerShell		
Backup-Features	Windows Server备份功能		
Backup		Windows Server备份	
Backup-Tools		用于备份的命 令行工具	
WSRM	Windows系统资源管理者		
WINS-Server	WINS服务器		
Wireless-Networking	无线局域网服务		

4.3.2 查询已安装的角色、角色服务与功能

在一个增强的命令提示符中,可以使用命令**servermanagercmd -query**来查询哪些角色、角色服务与功能已经安装到服务器中。该命令执行后,ServerManagerCmd会列出系统中每个可用的角色、角色服务与功能的配置状态。已安装的角色、角色服务与功能会突出显示并标记为已安装。在输出信息中,角色与角色服务会在功能前列出,如下面实例所示:

```

----- Roles -----
[ ] Active Directory Certificate Services [AD-Certificate]
    [ ] Certification Authority [ADCS-Cert-Authority]
    [ ] Certification Authority Web Enrollment [ADCS-Web-Enrollment]
    [ ] Online Responder [ADCS-Online-Cert]
    [ ] Network Device Enrollment Service [ADCS-Device-Enrollment]
[X] Active Directory Domain Services
    [X] Active Directory Domain Controller [ADDS-Domain-Controller]
    [ ] Identity Management for UNIX [ADDS-Identity-Mgmt]
    [ ] Server for Network Information Services [ADDS-NIS]
    [ ] Password Synchronization [ADDS-Password-Sync]
    [ ] Administration Tools [ADDS-IDMU-Tools]
...
----- Features -----
[ ] .NET Framework 3.0 Features [NET-Framework]
    [ ] .NET Framework 3.0 [NET-Framework-Core]
    [ ] XPS Viewer [NET-XPS-Viewer]
    [ ] WCF Activation [NET-Win-CFAC]
    [ ] HTTP Activation [NET-HTTP-Activation]
    [ ] Non-HTTP Activation [NET-Non-HTTP-Activ]
[X] BitLocker Drive Encryption [BitLocker]
[X] BITS Server Extensions [BITS]
[ ] Connection Manager Administration Kit [CMAK]
[X] Desktop Experience [Desktop-Experience]

```

除了用于粗略查看系统中已经安装了哪些组件之外,Servermanagercmd -query还可以用于记录服务器配置信息。要做到这一点,可以使用重定向符号(>)将该命令的输出保存在文本文件中,如下所示:

```
servermanagercmd -query > ServerConfig06-15-2008.txt
```

上面的命令中，将命令的输出信息保存到名为ServerConfig06-15-2008.txt的文本文件中。如果希望将结果保存到XML格式的文件中，只需要将-query后面的文件改为XML文件即可，比如：

```
servermanagercmd -query MySaveFile.xml
```

将输出结果保存在XML文件中的好处是，便于使用自动化技术进行处理。

4.3.3 安装角色、角色服务与功能

在一个增强的命令提示符中，可以通过命令**servermanagercmd -install *ComponentName***来安装角色、角色服务与功能。其中，*ComponentName*为所要安装的组件名（如表4-3、表4-4中所列出的）。通过包含-AllSubFeatures参数，还可以安装所有从属的组件，如下所示：

```
servermanagercmd -install fs-dfs -allsubfeatures
```

上面的命令中，在安装分布式文件系统角色服务的同时，也安装了从属于该角色服务的DFS Namespaces与DFS Replication等角色服务。成功安装后，其输出应该类似于如下的格式：

```
Start Installation...
[Installation] Succeeded: [File Services] Distributed File System.
[Installation] Succeeded: [File Services] DFS Namespaces.
[Installation] Succeeded: [File Services] DFS Replication.
<100/100>
```

```
Success: Installation succeeded.
```

如果需要重启来完成整个安装过程，则可以使用-Restart参数，借助于该参数，ServerManagerCmd可以重启计算机。如果需要在实际安装之前进行测试，则可以使用-Whatif参数。如果试图安装系统中已安装的组件，会得到如下的提示信息。其中声明“no changes were made”等内容，表明该组件已经安装：

```
NoChange: No changes were made because the roles, role services and features specified are already installed, or have already been removed from the local computer.
```

如果发生错误，并且ServerManagerCmd无法执行指定的操作，会得到错误信息。通常，错误消息文本以红色显示，并且包含了错误标志与错误文本，如下所示：

```
WriteError: Failed to write the log file: . Access to the path 'C:\Windows\logs\ServerManager.log' is denied.
```

上面的错误消息表明，ServerManagerCmd无法执行指定的操作，原因是无法获取对日志文件的写权限。安装组件时，ServerManagerCmd会将扩展的日志信息写入到%SystemRoot%\logs\servermanager.log这些扩展的日志信息详细记录了ServerManagerCmd的每一步操作。通过使用-Logpath参数或-L参数，也可以将这些信息写入到系统中其他位置。下面的实例中，将日志信息写入到c:\logs\install.log：

```
servermanagercmd -install fs-dfs -allsubfeatures
-logPath c:\logs\install.log
```

其他常见的错误还包括向命令行传递了无效参数等情况，如下面的错误信息所示：

```
ArgumentNotValid: Invalid parameters. Only specify either -install or
```

```
-remove.  
ArgumentNotValid: Invalid role, role service, or feature: 'fs-  
dfs'.The name was not found.
```

对于这种无效参数导致的错误,使用正确的参数即可解决。最后,还有一种错误是未以管理员身份运行命令提示符,此时会得到错误提示信息。其中声明ServerManagerCmd只能由本地计算机上内置管理员组中的某个成员运行,要解决这一错误,就需要以管理员权限来运行命令提示符。

4.3.4 移除角色、角色服务与功能

在一个增强的命令提示符中,可以通过命令**servermanagercmd -remove *ComponentName***来移除角色、角色服务与功能。其中,ComponentName为所要卸载的组件名(如表4-3、表4-4中所列出的)。通过包含-AllSubFeatures参数,还可以卸载所有从属的组件,如下所示:

```
servermanagercmd -remove fs-dfs -allsubfeatures
```

上面的命令中,在卸载分布式文件系统角色服务的同时,也卸载了从属于该角色服务的DFS命名空间与DFS复制等角色服务。成功卸载后,其输出应该类似于如下的格式:

```
Start Removal...  
[Removal] Succeeded: [File Services] Distributed File System.  
[Removal] Succeeded: [File Services] DFS Namespaces.  
[Removal] Succeeded: [File Services] DFS Replication.  
<100/100>
```

```
Success: Removal succeeded.
```

如果需要重启来完成整个卸载过程,则可以使用-Restart参数,借助于该参数,ServerManagerCmd可以重启计算机。如果需要在实际卸载之前进行测试,则可以使用-Whatif参数。如果试图卸载系统中未安装的组件,会得到如下的提示信息。其中声明no changes were made等内容,表明该组件尚未安装:

```
NoChange: No changes were made because the roles, role services and featur  
es specified are already installed, or have already been removed from the  
local computer.
```

如果发生错误,使得ServerManagerCmd无法执行指定的操作,会得到错误信息。卸载组件时,ServerManagerCmd会将扩展的日志信息写入到%SystemRoot%\logs\servermanager.log。与安装组件的过程类似,通过使用-Logpath参数或-L参数,也可以将这些信息写入到系统中其他位置。

作为系统管理员，主要的工作就是规划、组织系统与网络的正常运行，并能对网络运行的一些详细信息进行追踪与分析。如果你不想敷衍了事地应付这些工作，就应该学习如何以更加高效的方式来快速地完成这些工作。幸运的是，Windows提供了大量的命令行工具来帮助管理员完成这些任务，本章将讨论一些可用于系统日常管理的重要工具。

5.1 检查系统信息

通常，在使用某用户的计算机或远程服务器时，你可能需要检查一些基本的系统信息。比如，有哪些用户登录系统、当前的系统时间或者某特定文件在系统中的存放位置等。可用于收集基本系统信息的命令包括如下4个。

- **DATE**。可显示并设置当前系统日期。
- **TIME**。可显示并设置当前系统时间。
- **WHOAMI**。显示当前登录到系统的用户名，比如adatum\administrator。
- **WHERE**。使用某种搜索模式搜索文件，并返回一些匹配的结果。

要使用DATE或TIME命令，可以在命令shell窗口中键入命令，其后跟随/T参数，并按Enter键。比如，命令time/t的输出为**Wed 03/19/2008**，命令date/t的输出为**04:35 PM**。要设置系统日期与时间，则可以在相应命令后指定想要设置的日期或时间。设置当前日期时，输入日期的格式为MM-DD-YY。其中，MM、DD、YY分别代表用两个数字表示的月份、日期、年份。比如，要设置当前日期为2008年3月20日，则需要输入**03-20-08**。

设置当前时间时，输入时间的格式为HH:MM或HH:MM:SS。其中，HH、MM、SS分别为两个数字表示的小时、分数、秒数。如果输入时间时没有指定A.M.或P.M.，则系统会默认使用24小时制。其中，HH为00到11时代表A.M.，12到23时代表P.M.。下面几条命令的结果都是将时间设置为下午4点45分：

```
time 04:45 PM
time 04:45:00 PM
time 16:45:00
```

要使用WHOAMI命令来确定当前登录系统的用户，可以在命令shell窗口中键入该命令，之后按Enter键。如果该计算机是某工作组的一部分，则输出信息包括该计算机名，其后跟随一个反斜杠，之后跟随登录用户名，比如computer84\deanr。如果该计算机是某个域的一部分，则输出信息包括该计算机所属的域名，其后跟随一个反斜杠，之后跟随登录用户名，比如adatum\williams。

默认情况下，WHERE命令会在当前目录以及环境变量PATH指定的路径中进行搜索。因此，只要在命令shell窗口中键入where命令，其后跟随要搜索的可执行程序名，就可以快速地在当前路径中搜索该可执行程序。比如，要想搜索CMD.EXE，可以键入如下命令：

```
where cmd.exe
```

上面命令的输出结果为CMD.EXE的全文件路径：

```
C:\Windows\System32\cmd.exe
```

使用WHERE命令时，还有一种最常用的语法格式为：

```
where /r baseDir filename
```

上面的命令格式中，/r代表从指定的目录（\baseDir）开始递归搜索，包含所有子目录。filename代表要搜索文件的全名或部分名，其中可以包括通配符。?可以匹配单个字符，*可以匹配多个字符。比如，data???.txt或data*.*。下面的命令会在C:\目录及其所有子目录下搜索文件名以data开始的文本文件：

```
where /r C:\ data*.txt
```

下面的命令则搜索文件名以data开始的所有类型文件：

```
where /r C:\ data*.*
```

有时候，可能需要获取系统配置信息或系统环境信息。对于那些关键性的系统，可能还需要将这些信息保存或打印出来，用来检索和使用。如下一些命令可以帮助管理员收集系统信息。

- **DRIVERQUERY**。该命令可以显示一个列表，其中列出了系统中所有已安装的设备驱动程序及其属性，包括模块名、显示名、驱动程序类型、驱动程序链接日期等信息。使用详细模式进行输出时，该命令还会列出驱动程序状况、状态、启动模式、内存使用情况、文件系统路径等信息。通过使用/V参数，可以获取所有未签名驱动程序的详细输出。
- **SYSTEMINFO**。该命令可以显示详细的系统配置信息，包括操作系统版本、系统类型、系统制造商、处理器、BIOS版本、内存大小、系统区域设置、时区设置、网卡配置等信息。此外，该命令还可以显示系统中已经安装了哪些热点补丁。

要在本地系统中使用这些命令，可以在命令shell窗口中键入命令名，之后按回车。使用DRIVERQUERY命令时，可以指定/V参数来获取未签名驱动程序的详细信息，也可以指定/Si参数来显示签名驱动程序的属性信息，比如：

```
driverquery /si
```

使用DRIVERQUERY命令与SYSTEMINFO命令时，也可以指定要查询的远程计算机与运行许可权限。要做到这些，必须使用扩展的语法格式，包含如下参数：

```
/S Computer /U [Domain\] User [/P Password]
```

其中，Computer代表远程计算机名或IP地址，Domain代表域名（可选的，用户账号存在于该域内），User代表用户账号名（想使用的就是该用户的许可权限），Password代表该用户账号的口令（可选的）。如果不指定域名，则命令shell会将当前域名假定为该用户所在的域。如果不提供账号口令，则系统会呈现一个提示符要求输入口令。

要具体了解计算机与用户信息怎样被添加到这种扩展的语法格式中，可以参考如下的实例。

使用账号adatum\wrstanek查询MAILER1的驱动程序设置:

```
driverquery /s mailer1 /u adatum\wrstanek
```

使用账号adatum\administrator查询CORPSERVER01的系统信息:

```
systeminfo /s corpserver01 /u adatum\administrator
```

提示 这些命令的基本输出格式为表格格式。通过使用/Fo List或/Fo Csv, 也可以将输出格式分别转换为列表或以逗号分隔开的行。之所以要使用不同的输出格式, 是因为不同格式命令的输出信息存在较大差别。比如, 使用SYSTEMINFO命令时, 如果想查看关于系统配置信息的所有详细资料, 可以使用列表格式输出 (/Fo List); 使用DRIVERQUERY命令时, 如果正在对未签名驱动程序进行故障排除, 则可以使用详细列表格式输出 (/Fo List/V)。进一步地, 对于以后可能会导出到电子表格或非关系数据库的文件, 建议使用逗号分隔的行格式进行存储。此外, 通过使用输出重定向符号 (>或>>), 可以将DRIVERQUERY命令与SYSTEMINFO命令的输出信息重定向到特定的文件。

5.2 操作注册表

Windows注册表存储了配置信息, 通过使用命令行工具Reg, 可以很方便地对注册表键进行查看、添加、删除、比较、复制等操作。由于Windows注册表对Windows系统的正确运行是至关重要的, 在修改注册表之前, 要确保已经理解所做的修改对系统会有怎样的影响。在以任何方式编辑注册表之前, 应该进行完全的系统备份, 并创建系统恢复数据快照。这种做法的意义在于: 如果所做的修改是错误的, 或者对系统有不利影响, 就可以通过备份数据对系统与注册表进行恢复。

警告 错误修改Windows注册表可能会引发严重问题, 如果对注册表造成严重损坏, 你可能不得不重装系统。在执行注册表相关命令之前, 要认真理解该命令的用法, 确保这些命令能正确地达到预定执行目标。

5.2.1 理解注册表与键值

Windows注册表存储了操作系统、应用程序、用户以及硬件等设备的配置设置信息。这些信息是以注册表键与键值的形式存储的, 注册表键与键值存储在特定的root键下, 并由其控制不同键与键值的使用时间和方式。

表5-1列出了注册表的root键及其描述信息, 并给出了引用名(使用REG命令操作注册表时, 可以根据引用名来引用不同的root键)。在root键之下, 包含了一些子键, 用于对系统、用户、应用程序以及硬件的设置信息进行控制。这些子键在组织形式上为树结构, 使用文件夹来表示键值。比如, 在HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services键下, 可以发现一些与系统中已安装的所有服务相关的文件夹。这些文件夹中包含了注册表键值, 键值中存储了重要的服务配置设置信息以及一些子键。

要想操作某个注册表键, 必须指定其文件夹路径。比如, DNS键的路径为HKEY_LOCAL_

MACHINE\SYSTEM\CurrentControlSet\Services\DNS，要查看并操作该键，可以使用缩略路径名 HKLM\SYSTEM\CurrentControlSet\Services\DNS。

表5-1 Windows注册表中的键

root键	引用名	描述
HKEY_CURRENT_USER	HKCU	存储了当前用户的配置设置信息
HKEY_LOCAL_MACHINE	HKLM	存储了系统级的配置设置信息
HKEY_CLASSES_ROOT	HKCR	存储了应用程序与文件的配置设置信息，可以确保系统使用正确的应用程序打开要访问的文件
HKEY_USERS	HKU	存储了默认用户与其他用户的配置设置信息（根据配置文件）
HKEY_CURRENT_CONFIG	HKCC	存储了使用中的硬件配置文件的相关信息

注册表键值以一些特定的数据类型进行存储，表5-2总结了注册表键值使用的一些主要数据类型。

表5-2 注册表键值及其数据类型

数据类型	描述	实例
REG_BINARY	用于识别二进制值。二进制值采用二进制形式存储（0、1 组合），但在显示与输入时采用十六进制格式	01 00 14 80 90 00 00 9c 00
REG_DWORD	用于识别二进制数据类型，以4字节长度的十六进制格式存储32位的整数值	0x00000002
REG_EXPAND_SZ	用于识别可扩展的字符串值，通常用于存储目录路径	%SystemRoot%\dns.exe
REG_MULTI_SZ	用于识别多个字符串值	Tcpip Afd RpcSc
REG_NONE	用于识别没有指定特定数据类型的数据。这种数据采用二进制形式存储，但在显示与输入时采用十六进制格式	23 45 67 80
REG_SZ	用于识别包含字符序列的字符串值	DNS Server

在知道了注册表键路径以及可用的键值数据类型之后，就可以以多种方式使用REG命令对注册表键进行查阅和操作。REG命令有一些不同的子命令，我们将在下面分别讨论如下的一些子命令。

- REG add。为注册表添加一个新子键或条目。
- REG delete。从注册表删除一个子键或条目。
- REG query。列出某注册表键下的条目以及子键名（如果存在）。
- REG compare。比较注册表键或条目。
- REG copy。将注册表条目复制到特定的注册表键路径（本地或远程系统上）。
- REG flags。显示并管理指定键的当前标记。
- REG restore。将保存的子键、条目、键值等写回到注册表。
- REG save。将指定的子键、条目、键值保存到文件。

除上述命令之外，还对如下一些命令进行了讨论，这些命令可以执行一些高级的注册表操作功能。

- REG import。将指定的hive文件导入到注册表。
- REG export。将指定的子键、条目、键值导出到注册表文件。
- REG load。将指定的hive文件加载到注册表。
- REG unload。从注册表卸载指定的hive文件。

注解 REG命令以当前用户的权限运行。如果想使用不同的权限，最简单的方式是以具有该权限的用户账号登录。

5.2.2 查询注册表值

使用REG query命令，可以引用想要操作的键名或键值的全路径来读取注册表值，其基本语法为：

```
reg query KeyName [/v ValueName]
```

其中，*KeyName*为要检查键的键名，*ValueName*是一个可选的参数，用于指定特定的键值。下面给出的实例中，对current control set中的DNS键进行了查询：

```
reg query HKLM\SYSTEM\CurrentControlSet\Services\DNS
```

对于这种查询，如果知道待检查键的特定键值，还可以使用另一种方法，该方法使用/V参数对查询结果进行限定。下面的实例中，列出了DNS键的ImagePath条目的值：

```
reg query HKLM\SYSTEM\CurrentControlSet\Services\DNS /v ImagePath
```

注册表键路径也可以包含UNC名或远程计算机的IP地址，比如\\Mailer或\\192.168.1.100。不过，对远程计算机，只能对HKLM与HKU这两个root键进行操作。下面的实例中，对MAILER1上的DNS键进行查询：

```
reg query \\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS
```

注解 如果指定了不存在的键或键值，命令shell会显示一条错误消息。典型情况下，该消息为错误：系统找不到指定的注册表项或值。

5.2.3 比较注册表值

使用REG compare命令，可以对两台系统上相同的注册表条目与键进行比较，也可以对同一系统上两个不同的键进行比较。在如下的场景下，进行注册表比较是有用的。

- 对服务与应用程序配置的问题进行故障排除。这种情况下，对两台不同系统的注册表配置进行比较是有用的。理想情况下，两台系统中应该有一台进行了正确配置，要检查的那台配置失误，通过对注册表中相关配置键的比较，就可以发现问题所在。
- 确保应用程序或服务在多台系统上进行了相同配置。这种情况下，可以使用一台系统作为基准，以便对其他系统的配置进行测试和调整。理想情况下，基准系统应该先按需求目标进行正确配置，之后在对其他系统的配置进行比较和调整。

REG compare命令的基本语法格式为：

```
reg compare KeyName1 KeyName2 [/v ValueName]
```

其中，*KeyName1*与*KeyName2*为待比较的子键名，键名可以包含UNC名或待检查远程计算机的IP地址。*ValueName*为可选的参数，用于指定待比较的特定键值。下面的实例中，对MAILER1与MAILER2上current control set下的DNS键进行比较：

```
reg compare \\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS
```



```
\\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS
```

如果这两台系统上该键的配置相同, 则输出信息为:

```
Results Compared: Identical
The operation completed successfully.
```

如果这两台系统上该键的配置不同, 则输出信息会显示这种差别。以<字符开始的差别信息属于第一个键, 以>字符开始的差别信息则属于第二个键。此外, 输出信息中还包含如下声明:

```
Results Compared: Different
The operation completed successfully.
```

提示 之所以会显示差别信息, 是因为/Od参数是默认使用的。通过指定其他参数, 还可以在输出信息中查看所有的差别信息与匹配信息(/Oa)、只查看匹配信息(/Os), 或者没有结果信息(/On)。

此外, 如果需要递归地比较所有子键与条目, 可以使用/S参数, 如下所示:

```
reg compare \\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS
\\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS /s
```

通过这一命令, 可以对MAILER1与MAILER2上DNS键及其所有子键与相关条目进行比较。

5.2.4 注册表键的保存与恢复

在对注册表条目进行修改之前, 较好的做法是对要修改的键进行保存。这样, 如果有任何错误发生, 都可以将这些键恢复到原始的设置。要保存注册表键及其相关子键与值, 可以使用REG save命令, 如下所示:

```
reg save KeyName "FileName"
```

其中, *KeyName*为要保存的子键的路径, *FileName*为要创建的注册表hive文件名。子键路径也可以包含UNC名或远程计算机的IP地址, 不过, 对远程计算机, 只能对HKLM与HKU这两个root键进行操作。此外, 文件名应该包含在双引号中, 并以.hiv扩展名结尾, 以表明其为注册表hive文件, 如下面实例所示:

```
reg save HKLM\SYSTEM\CurrentControlSet\Services\DNS "DNSKey.hiv"
```

通过这一命令, 将DNS子键及其相关子键与值保存到名为Dnskey.hiv的文件中。文件名中也可以包含目录路径, 如下所示:

```
reg save \HKLM\SYSTEM\CurrentControlSet\Services\DNS
"\\Mailer1\SavedData\DNSKey.hiv"
```

如果注册表hive文件存在, 则命令shell会给出提示信息询问是否重写文件, 此时选择Y, 也即重写文件。如果希望默认地重写文件, 而不弹出提示信息, 则可以使用/Y参数。

要恢复以前保存的注册表键, 可以使用Reg restore命令, 该命令的语法格式为:

```
reg restore KeyName "FileName"
```

其中, *KeyName*为要恢复的子键的路径, *FileName*为用于恢复注册表键的原始注册表hive文件名。与REG copy命令不同的是, Reg restore命令只能在本机执行, 而不能使用该命令对远程计算机上的注册表键进行恢复。不过, 你可以在远程计算机上启动一个远程桌面会话, 之后通过远程桌面登录

来恢复本地机上的注册表键。

下面给出了一个使用REG restore命令的实例：

```
reg restore HKLM\SYSTEM\CurrentControlSet\Services\DNS "DNSKey.hiv"
```

该命令对以前保存在DNSKey.hiv文件中的DNS键进行了恢复。

5.2.5 添加注册表键

要向Windows注册表中添加子键与值，可以使用REG add命令，其基本语法为：

```
reg add KeyName /v ValueName /t DataType /d Data
```

其中，*KeyName*为所要使用的注册表键的键名，*ValueName*为所要创建的子键或键值，*DataType*为数据类型，*Data*为要插入的实际值。尽管这一命令似乎涉及到很多数据和对象，但实际上该命令的使用是很直接的，参考如下实例：

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\DNS /v DisplayName  
/t REG_SZ /d "DNS Server"
```

上面命令的功能是：向注册表中的DNS键添加一个名为DisplayName的键值，该键的入口为一个包含有DNS Server的字符串。该命令中使用了双引号，因为字符串中包含了空格，使用双引号才能保证命令shell对字符串的正确理解。如果要添加的键或值已经存在，命令shell会给出提示信息询问是否重写数据。选择Y重写，选择N则放弃执行。如果希望默认地重写文件，而不弹出提示信息，则可以使用/F参数。

设置可扩展的字符串值时（REG_EXPAND_SZ），必须使用插入记号（^）符号对百分号（%）进行转义处理。这里，%用于指定环境变量。参考如下实例：

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\DNS /v ImagePath  
/t REG_EXPAND_SZ /d ^%SystemRoot%\System32\dns.exe
```

其中，使用了`^%SystemRoot^%`，以便确保环境变量SystemRoot的正确输入以及命令shell的正确理解。

设置非字符串值时，不需要使用引号，如下所示：

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\DNS /v ErrorControl  
/t REG_DWORD /d 0x00000001
```

5.2.6 复制注册表键

使用REG copy命令，可以将注册表条目复制到本地机或远程系统上的某一位置。该命令的基本语法为：

```
reg copy KeyName1 KeyName2
```

其中，*KeyName1*为要复制的源子键路径，*KeyName2*为目标子键路径。需要说明的是，尽管子键路径可以包含UNC名或远程计算机的IP地址，但使用REG copy命令对远程的源注册表键与目标注册表键进行操作时，仍然存在一些限制，如下所示。

- ❑ 远程的源注册表子键只能使用HKLM与HKU这两个root键。
- ❑ 远程的目标注册表子键只能使用HKLM与HKU这两个root键。

下面的实例中，将本地系统上的DNS子键复制到MAILER2上的DNS子键：

```
reg copy HKLM\SYSTEM\CurrentControlSet\Services\DNS
\\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS
```

通过使用/S参数,可以复制指定的子键以及该子键下的所有子键与注册表条目。下面的实例中,复制了DNS子键及其相关的所有子键与值:

```
reg copy HKLM\SYSTEM\CurrentControlSet\Services\DNS
\\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS /s
```

如果目标路径中已经存在要复制的值,则REG copy命令执行时会弹出提示信息,对每个已存在的值都需要确认是否重写。你可以根据实际情况选择Y或N,也可以按A键来重写所有已存在的值而不再显示提示信息。

注解 如果不希望看到提示信息,可以使用/F参数重写所有已存在的值而不显示提示信息。不过,在重写已存在的注册表键之前,你可能需要对其进行保存,以便在发生问题的时候对其进行恢复。为此,可以使用5.2.4中讨论的REG save命令与REG restore命令。

5.2.7 删除注册表键

要从Windows注册表中删除子键与值,可以使用REG delete命令。REG delete命令有几种不同的语法格式,如果需要删除某子键以及该子键下所有子键与注册表条目,可以使用如下语法:

```
reg delete KeyName
```

其中,KeyName为所要删除的子键名。此外,尽管子键路径可以包含UNC名或远程计算机的IP地址,但远程源子键只能使用HKLM与HKU这两个root键,参考如下实例:

```
reg delete \\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS2
```

该命令删除了MAILER1上的DNS2子键以及该子键下的所有子键与条目。

如果需要限定删除的范围,比如,只删除子键下某一个特定的条目,则可以使用如下的语法:

```
reg delete KeyName /v ValueName
```

其中,KeyName为所要处理的子键名,ValueName为所要删除的特定的条目名。与前面类似,这里的子键路径也可以包含UNC名或远程计算机的IP地址,但远程源子键只能使用HKLM与HKU这两个root键。下面的实例中,删除了MAILER2上DNS2子键的Description条目:

```
reg delete \\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS2 /v
Description
```

提示 对上面两种情况,命令shell都会给出提示信息,询问是否永久性地删除指定的注册表条目。如果是则选择Y。也可以使用/F参数来默认删除,而不再弹出提示信息。另外一个有用的参数是/Va,该参数的作用是,只删除子键下的值,而不删除该子键下的其他子键。

5.2.8 导入与导出注册表键

有时候,可能需要将注册表的全部或部分复制到一个文件中,之后在其他计算机上使用该文件。比如,在计算机1上安装了一个组件,并对其进行了一些扩展的配置,之后需要在计算机2上也使用该

组件，但不希望重复整个配置过程。要做到这一点，可以在计算机2上安装该组件，对其进行初始的配置，之后从计算机1中导出该组件的注册表配置文件，并将该文件导入到计算机2中，从而保证该组件的正确配置，又避免了低效的重复。当然，这种方法只适用于组件的完整配置设置存储在注册表中的情况，但从中可以看出这种注册表导出、导入的作用和能力。

使用REG export命令与REG import命令时，导出、导入注册表数据是非常容易的，包括特定的root键的分支数据以及个别的子键及其包含的数据。导出数据时，要创建一个.reg文件来保存导出的注册表数据。该文件实际上是一个脚本文件，通过REG import命令，可以将其导入到原计算机或其他计算机上的注册表中。

由于注册表文件是以标准的文本写入的，因此可以使用任意的标准文本编辑器对其进行查阅与修改（如果需要）。要将注册表数据导入到当前目录下的文件中，可以使用如下的语法：

```
reg export KeyName FileName
```

其中，*KeyName*为要使用的子键名，*FileName*为用于存储导出的注册表数据的文件名。与前面类似，这里的子键路径也可以包含UNC名或远程计算机的IP地址，但远程源子键只能使用HKLM与HKU这两个root键。下面的实例中，导出了MAILER1上的MSDTC子键：

```
reg export \\Mailer1\HKLM\SOFTWARE\Microsoft\MSDTC msdtc-regkey.reg
```

通过REG export命令，可以导出任意层次的注册表数据。比如，如下命令可以导出HKLM root键及其所有子键：

```
reg export HKLM hklm.reg
```

提示 通过/Y参数，可以使REG export命令在导出注册表数据时默认重写已存在的文件。通过regedit /e SaveFile命令，可以导出完整的注册表数据。其中，SaveFile为所要保存的注册表文件存储位置的全路径。比如，如果需要将整个注册表数据复制到C:\Save\Regdata.reg，可以使用命令regedit /e C:\Save\Regdata.reg。

导入注册表数据会将注册表脚本文件的内容添加到目标计算机的注册表中，添加数据时，或者是创建新的注册表键与值（如果此前不存在），或者是重写注册表键与值（如果此前存在）。以如下的语法格式使用REG import命令，可以导入注册表数据：

```
reg import FileName
```

其中，*FileName*为当前目录中要导入的注册表文件名，比如：

```
reg import msdtc-regkey.reg
```

要注意的是，不能远程执行导入命令，也不能导入非本地的注册表文件。导入注册表键时，必须登录本地系统，并且要导入的注册表文件也必须存在于本地计算机上。

5.2.9 加载与卸载注册表键

正如有时候需要导出或导入注册表数据一样，有时候你还需要使用单独的注册表hive文件。最可能导致此需求的原因是：必须修改某用户的配置文件，以便纠正其中存在的、导致该用户无法访问或使用系统的问题。比如，由于某用户不恰当地改变了显示模式，导致其变为一种无效的设置，因此无法访问该计算机，这时候就需要加载并修改该用户配置文件的设置。将该用户的配置文件数据

设置加载到注册表中后,就可以编辑注册表,纠正其中存在的问题,并保存所做的改变,之后该用户就可以登录系统。

另一个需要加载注册表键的原因是:需要改变远程系统上注册表中某一个特定的部分。需要说明的是,加载与卸载注册表hive文件只影响HKEY_LOCAL_MACHINE与HKEY_USERS,只有在选择了其中的某一个时,才能执行加载与卸载操作。加载注册表hive文件时,加载数据并不会替换选定的root键,而是变为该root键的子键。由于HKEY_LOCAL_MACHINE与HKEY_USERS是构建系统注册表中所有root键的基础,因此,实际上这两个命令可以操作注册表中的任何区域。

进行加载操作之前,必须使用REG save命令将要加载的注册表数据保存为注册表hive文件。以如下语法使用REG load命令,可以将以前保存的注册表hive文件加载到注册表中:

```
reg load RootKey\KeyName FileName
```

其中,RootKey为加载命令使用的root键(加载的临时注册表键将在其下创建),KeyName为所要创建的临时子键的名,FileName为所要加载的注册表hive文件名。要注意的是,只能在HKLM或HKU这两个root键下创建临时子键。下面的实例中,在HKLM下创建了一个名为CurrTemp的临时子键,并将Working.hiv文件加载到该子键中:

```
reg load HKLM\CurrTemp Working.hiv
```

再次重申:不能远程执行导入命令,也不能导入非本地的注册表文件。导入注册表键时,必须登录本地系统,并且要导入的注册表文件也必须存在于本地计算机上。

加载了注册表键之后,就可以使用前面讨论的技术对其子键与值进行操作。在完成了对某注册表键的修改后,可以使用REG save命令将其保存到新的注册表文件中。保存后,就可以以如下的语法格式使用REG unload命令,以便卸载hive文件,并将其从内存与注册表中移除:

```
reg unload RootKey\KeyName
```

其中,RootKey为卸载命令使用的root键(临时注册表键在其下创建),KeyName为所要卸载的临时子键的名。下面的实例中,从HKLM下卸载了一个名为CurrTemp的临时子键,并将Working.hiv文件加载到该子键中:

```
reg unload HKLM\CurrTemp
```

注解 不能对正在被操作系统或其他进程使用的hive文件本身进行卸载等操作,不过可以复制一个该文件的副本,之后对文件副本进行处理。要保存文件副本,可以使用reg save命令,其后跟随要保存的root键的缩略名以及该hive文件的文件名。比如,命令reg save hkcu c:\currhkcu.hiv可以将HKEY_LOCAL_MACHINE保存到名为Currhkcu.hiv的hive文件中,该文件存在于C盘根目录。尽管可以以这种模式保存注册表的逻辑root键(HKCC、HKCR、HKCU),但本技术中只能保存HKLM、HKU下的子键。

如果需要修复远程计算机上注册表的某个区域,可以遵循如下规则与步骤。

- (1) 访问远程计算机,使用REG save命令将注册表hive保存到文件中。
- (2) 使用XCOPY命令或类似命令将注册表文件复制到本地系统。
- (3) 使用REG load命令,将相关的hive文件复制到本地机的注册表中。
- (4) 进行必要的修改,之后使用REG save命令保存所做的修改。

(5) 将修改后的hive文件上传到远程计算机，之后使用REG import命令将其导入注册表中，以便修复存在的问题。

(6) 在远程计算机上测试有效后，使用REG unload命令从本地机上卸载hive文件。

5.3 管理系统服务

系统服务为工作站与服务器提供了关键功能。要对本地机与远程计算机上的系统服务进行管理，可以使用服务控制器命令SC，该命令包含很多子命令，这里只讲述其中的一部分。具体地说，接下来将逐一讨论如下一些命令。

- SC config。用于配置服务启动与登录账号。
- SC query。用于显示计算机上已配置的所有服务的列表。
- SC qc。用于显示某特定服务的配置。
- SC start。用于启动服务。
- SC stop。用于结束服务。
- SC pause。用于暂停服务。
- SC continue。用于恢复服务。
- SC failure。用于设置服务失效时执行的操作。
- SC qfailure。用于查看服务失效时执行的操作。

上面这些命令都可以对远程计算机的服务进行操作，方法是在要使用的子命令前插入UNC名或远程计算机的IP地址，其语法格式为：

```
sc ServerName Subcommand
```

5.3.1 查看已配置的服务

要查看系统中已配置的所有服务，可以在命令提示符中键入如下命令：

```
sc query type= service state= all
```

或

```
sc ServerName query type= service state= all
```

其中，*ServerName*为UNC名或远程计算机的IP地址，比如\\Mailer1或\\192.168.1.100，如下面的实例所示：

```
sc \\Mailer1 query type= service state= all
sc \\192.168.1.100 query type= service state= all
```

注解 在type= service与state= all中使用的等号(=)后面，必须紧跟着一个空格。如果没有使用空格，则命令会失败。

对state标记，可以赋值为active（只显示运行中的服务），也可以赋值为inactive（显示所有暂停的或终止的服务），如下所示：

```
sc \\Mailer1 query type= service state= active
sc \\Mailer1 query type= service state= inactive
```

上面第一个实例中，查询MAILER1上所有运行中的服务，并返回一个列表。第二个实例中，查询MAILER1上所有终止状态的服务，并返回一个列表。

SC query命令的输出信息展示了各种服务及其配置信息，每个服务条目的格式如下：

```
SERVICE_NAME: W3SVC
DISPLAY_NAME: World Wide Web Publishing Service
      TYPE           : 20  WIN32_SHARE_PROCESS
      STATE          : 4   RUNNING
                        (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
      WIN32_EXIT_CODE : 0   (0x0)
      SERVICE_EXIT_CODE : 0   (0x0)
      CHECKPOINT      : 0x0
      WAIT_HINT       : 0x0
```

作为管理员，下面是操作最多的字段。

- **Service Name**。服务的缩略名，只有系统中已安装的服务才会列出，如果需要的服务没有列出，则需要安装该服务。
- **Display Name**。描述性的服务名。
- **State**。服务所处的状态（运行、暂停、终止）。

上面可以看出，SC query命令的输出信息非常之多，要想只获取自己关注的信息，可以在该命令中使用过滤机制。比如，使用如下命令，可以对输出信息进行过滤，使其只显示最重要的字段：

```
sc query type= service 1 find /v "x0"
```

上面的命令行中，使用FIND命令对SC query命令的输出进行了过滤，使得其输出为如下的形式：

```
SERVICE_NAME: W3SVC
DISPLAY_NAME: World Wide Web Publishing Service
      TYPE           : 20  WIN32_SHARE_PROCESS
      STATE          : 4   RUNNING
                        (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
```

注解 参数/V " x0" 的作用是，使得FIND命令只显示那些不包括文字x0的输出行，x0是WIN32_Exit_Code、Service_Exit_Code、Checkpoint、Wait_Hint等字段中常见的文本。通过这种方法，就在显示的输出信息中去除了不需要的字段。

如果已知要使用的服务名，就可以使用SC qc命令来显示其配置信息，其语法为：

```
sc qc ServiceName
```

其中，ServiceName为所要处理的服务名，这种格式命令的输出信息类似于：

```
SERVICE_NAME: w3svc
      TYPE           : 20  WIN32_SHARE_PROCESS
      START_TYPE      : 2   AUTO_START
      ERROR_CONTROL   : 1   NORMAL
      BINARY_PATH_NAME : C:\WINDOWS\System32\svchost.exe -k iissvcs
      LOAD_ORDER_GROUP :
      TAG             : 0
      DISPLAY_NAME     : World Wide Web Publishing Service
      DEPENDENCIES     : RPCSS
                      : HTTPFilter
```

```

: IISADMIN
SERVICE_START_NAME : LocalSystem

```

可以看出，输出信息中没有包含服务的当前状态，但包含了如下一些与该服务相关的要素。

- 二进制路径名。该服务的可执行文件路径。
- 依赖关系。该服务正常运行的一些依赖因素。
- 显示名。描述性的服务名。
- 服务启动名。该服务的登录账号名。
- 启动类型。该服务的启动配置。

注解 配置为自动启动的服务的启动类型为AUTO_START，配置为手动启动的服务的启动类型为DEMAND_START，已禁用服务的启动类型为DISABLED。

- 类型。服务类型及其是否为共享进程。

注解 配置服务登录时，有些情况下，需要了解进程是在自己专有的上下文中运行，还是与其他进程共享运行。共享进程在类型中列出为WIN32_SHARE_PROCESS，在自己专有上下文中运行的进程在类型中列出为WIN32_OWN_PROCESS。

5.3.2 启动、终止与暂停服务

作为管理员，经常需要对Windows服务进行启动、终止或暂停等操作，相关的命令及其语法分别为：

启动某服务

```
sc start ServiceName
```

暂停某服务

```
sc pause ServiceName
```

恢复暂停的服务

```
sc continue ServiceName
```

终止某服务

```
sc stop ServiceName
```

上述命令中，*ServiceName*均指待处理的服务的缩略名，比如：

```
sc start w3svc
```

在所有的SC命令中，都可以通过指定远程计算机名来使用其中的服务。比如，要启动MAILER1上的w3svc服务，可以使用如下的命令：

```
sc \\Mailer1 start w3svc
```

上面命令输出结果中状态应该为STATE_PENDING，对于stop、pause、continue等命令，状态则分

别对应STOP_PENDING、PAUSE_PENDING、CONTINUE_PENDING。如果得到错误的结果，则输出信息中会声明FAILED，并给出错误文本，其中详细描述了导致失败的原因。比如，如果试图启动一项已经启动的服务，则会得到错误信息：

```
An instance of the service is already running.
```

如果试图终止一项已经终止的服务，则会得到如下的错误信息：

```
The service has not been started.
```

5.3.3 配置服务的启动方式

对Windows服务，可以将其设置为手动启动或自动启动，也可以通过对服务的禁用来永久性地关闭服务。通过如下命令，可以配置服务的启动：

```
sc config ServiceName start= flag
```

其中，*ServiceName*为所要使用的服务的缩略名，*flag*为所要使用的启动类型。对Windows服务而言，有效的*flag*值包括如下几个。

- ❑ **Auto**。在系统启动时启动服务。
- ❑ **Demand**。根据需要手工启动服务。
- ❑ **Disabled**。禁用服务。
- ❑ **Delayed-Auto**。延迟服务的启动，直至所有非延迟的自动服务都已启动。

根据这些*flag*值的含义，要将某服务配置为自动启动，可以使用命令：

```
sc config w3svc start= auto
```

或

```
sc \\Mailer1 config w3svc start= auto
```

注解 在start=auto中使用的等号(=)后面，必须紧跟着一个空格。如果没有使用空格，则命令会失败。另外，上面的命令只会报告SUCCESS或FAILURE，而不会报告说服务已经在指定的启动模式中进行了配置。

禁用某服务并不会立即终止该服务，而是在计算机下次启动的时候不再启动该服务。要确保某服务被禁用并且终止，可以在使用SC stop命令后再运行SC config命令。

5.3.4 配置服务的登录方式

对Windows服务，可以将其配置为以系统账号登录，也可以将其配置为以特定用户账号登录。比如，要使得某服务以本地系统账号登录，可以使用如下命令：

```
sc config ServiceName obj= LocalSystem
```

其中，*ServiceName*为正在配置的、准备以本地系统账号登录的服务名。如果某服务提供了一个可操作的用户界面，就可以在其中添加标记 **type=interact type=own**，如下面实例所示：

```
sc config w3svc obj= LocalSystem type= interact type= own
```

其中，*type=interact*标记表明该服务可以与Windows桌面进行交互，*type= own*标记表明该服务在自己

专有的进程空间运行。在某服务与其他服务共享其可执行文件的情况下，则应该使用标记`type=share`，如下面实例所示：

```
sc config w3svc obj= LocalSystem type= interact type= share
```

提示 如果不知道某服务是以共享进程的方式运行，还是以自己专有进程空间运行，则可以使用SC qc命令查询该服务的启动类型。SC qc命令在5.3.1中进行了讨论。

服务也可以配置为以指定的账号登录，要做到这一点，可以使用如下命令：

```
sc config ServiceName obj= [Domain\]User password= Password
```

其中，*Domain*为可选的域名（要指定的用户账号存在于该域中），*User*为要使用的用户账号名，*Password*为该账号的口令。参考如下实例：

```
sc config w3svc obj= adatum\webbies password= blue5!CraZy
```

上面的命令中，将W3svc服务配置为使用Adatum域中的Webbies账号登录。该命令应该在输出信息中声明SUCCESS或FAILURE。如果账号名无效或者不存在，或者提供的口令无效，则命令会失败。

注解 如果某服务配置为在本地系统账号下与桌面交互，则不能将其改变为在域账号下运行，除非使用标记`type=own`，此时的语法格式为：`sc config ServiceName obj= [Domain\] User password= Password type=own`。

真实场景 作为管理员，应该保持对那些由服务使用的账号的追踪和审计。如果没有进行正确配置，这些账号就可能成为重大安全问题的根源。对这些由服务使用的账号，应该采用最严格的安全设置，并在保证其完成必要功能的前提下赋予其尽可能小的权限。典型情况下，由服务使用的账号并不需要很多通常用户账号所需要的权限。比如，大多数服务账号不需要本地登录的权限。管理员应该准确了解系统中使用了哪些服务账号（以便于对其进行追踪和审计），并且应该向配置管理员账号一样对其进行配置。当然，这并不是说赋予其管理员权限，而是说应该为其设置安全的口令，对账号的使用进行认真的监测，以及对账号权限与特权的小心使用等。

5.3.5 配置服务的恢复方式

使用SC failure命令，可以将Windows服务配置为在服务失败时采取特定的动作。比如，可以在服务失败时重启该服务或者运行某个应用程序。

你可以为服务恢复选项设置第一个、第二个以及依次类推的恢复尝试，每次服务失败发生时，当前的失败计数会递增。你也可以设置一个参数，用于规定失败计数器重置之前必须消逝的时间长度。比如，可以规定，在最后一次失败发生24小时之后重置失败计数器。

在对服务恢复进行配置之前，可以使用SC qfailure命令检查当前的服务恢复设置，其语法为：

```
sc qfailure ServiceName
```

其中，*ServiceName*为所要设置的服务名，比如：

```
sc qfailure w3svc
```


此外,也可以对远程计算机进行检查,比如:

```
sc \\Mailer1 qfailure w3svc
```

或

```
sc \\192.168.1.100 qfailure w3svc
```

这些命令的输出信息中,失败动作是以前应该执行的先后顺序列出的。在下面的实例输出中,W3svc服务的恢复策略设置为:服务第一次、第二次失败时都尝试重启该服务,在第三次失败时则重启计算机。

```
[SC] QueryServiceConfig2 SUCCESS
```

```
SERVICE_NAME: w3svc
  RESET_PERIOD (in seconds) : 86400
  REBOOT_MESSAGE           :
  COMMAND_LINE             :
  FAILURE_ACTIONS          : RESTART -- Delay = 1 milliseconds.
                           : RESTART -- Delay = 1 milliseconds.
                           : REBOOT -- Delay = 1000 milliseconds.
```

注解 对某些关键性的系统服务,Windows会自动地为其配置恢复策略。典型情况下,这些服务的恢复策略设置为在失败时重启该服务。还有一些服务配置为失败时运行某程序,比如,IIS Admin服务配置为失败时运行一个名为lisreset.exe的程序,该程序可以在重启IIS Admin服务的同时纠正服务中存在的问题,并安全地管理IIS服务。

用于配置服务恢复的命令是SC failure,其基本语法为:

```
sc failure ServiceName reset= FailureResetPeriod actions= RecoveryActions
```

其中,ServiceName为所要配置的服务名,FailureResetPeriod规定了在重置失败计数器之前应该达到的正常运行时间(以秒为计数单位),RecoveryActions为失败发生时执行的操作以及该操作启动之前的延迟时间(以毫秒为计数单位),可用的恢复操作如下所示。

- 无操作(以空字符串" "为标志)。操作系统不会为本次失败尝试恢复操作,但可能为此前或此后的其他失败进行恢复操作。
- 重启服务。终止该服务,并在短暂暂停后重启该服务。
- 运行一个程序。失败时运行一个程序或脚本,脚本可以是一个批处理脚本,也可以是一个Windows脚本。如果选择这个恢复操作,要注意为所要运行的程序设置完整的文件路径,并设置程序启动时需要传递给该程序的命令行参数。
- 重新引导计算机。关闭计算机,并在指定的延迟时间达到后重启计算机。

最佳实践 为关键性的系统服务设置恢复选项时,可以采用的一个较好策略是:在第一次、第二次失败时重启该服务,第三次失败时则重新引导服务器。

使用SC failure命令时,要记住如下几个要点。

- 重置时间间隔的计数单位为秒。重置计数器的时间间隔通常设置为几个小时或几天,但计数单位是秒。一个小时有3,600秒,一天有86,400秒。比如,要设置两小时的重置时间间隔,应

该使用的值为7,200。

- 每一个恢复操作后必须跟随执行该操作之前需要等待的时间（计数单位为毫秒）。对于重启服务的操作，可能需要较短的延迟，比如1毫秒（无延迟），1秒（1,000毫秒），或者5秒（5,000毫秒）。对于重启计算机的操作，则可能需要较长的延迟，比如15秒（15,000毫秒），或30秒（30,000毫秒）。
- 在输入操作及其延迟时间时，应该将其作为一个统一的文本字符串，每个值之间都以正斜杠（/）间隔。比如，可以使用如下的形式：restart/1000/restart/1000/reboot/15000，其含义为：第一次、第二次失败时，在1秒的延迟后重启该服务，第三次失败时，在15秒的延迟后重新引导计算机。

参考如下实例：

```
sc failure w3svc reset= 86400 actions= restart/1/restart/1/reboot/30000
```

根据上面命令行中的约定，第一次、第二次服务失败时，几乎立即重启该服务，第三次失败时，则在30秒的延迟后重新引导计算机。此外，如果在24小时（86,400秒）的时间间隔内没有发生服务失败，则重置失败计数器。与前面实例中类似，你也可以在上面的命令行中指定远程计算机（插入UNC名或远程计算机的IP地址）。

如果要在服务失败时使用Run操作，则可以使用`Command= parameter`来指定要运行的命令或程序，`parameter`负责指定要运行的命令的全路径以及传递给该命令的参数。要注意的是，命令路径与参数等内容需要使用双引号封装在一起，如下面实例所示：

```
sc failure w3svc reset= 86400 actions= restart/1/restart/1/run/30000  
command= "c : \restart_w3svc.exe 15"
```

5.4 从命令行重启与关闭系统

关机或重启系统是经常要执行的系统管理任务。要完成这些任务，一种方法是使用Shutdown工具，该工具可同时用于本地机与远程系统。另一种方法是对关机进行调度，比如，使用Schtasks指定关机的执行时间，或者创建一个脚本，其中包含对某台系统的关机命令列表。

真实场景 尽管大多数情况下Windows系统的启动与关机都不会遇到问题，但偶尔也可能会在执行过程中停止响应。如果出现这种情况，应该设法找到导致问题的根源以便于故障排除，下面给出了几种可能导致系统停止响应的原因。

(1) 系统正在尝试执行（或正在运行）一个启动脚本或关机脚本，该脚本尚未执行完毕，或者脚本本身失去响应（在这种情况下，系统需要等待直至脚本超时）。

(2) 启动初始化文件或服务导致的问题，如果确认，就需要使用系统配置工具（Msconfig）对启动项进行故障排除。禁用服务、禁用某启动项或启动初始化文件中的某个条目也可能解决问题。

(3) 系统中安装的防病毒程序导致的问题。比如，有些情况下，在关机时，防病毒程序会扫描可移除的介质驱动器。为解决这一问题，可以对防病毒程序进行配置，使其在系统关机时不再扫描可移除的介质驱动器或其他带有可移除介质的驱动器，也可以临时性地禁用或关闭防病毒程序。

(4) 错误地配置了声音设备导致的启动与关机问题。为确定可能的问题根源，可以依次检查这些设备。关闭声音设备，之后重启计算机。如果问题解决，说明确实是声音设备的问题，则需要为声音

设备安装新的驱动程序（另外一种可能性不是很大的原因是开机、关机的声音文件损坏）。

(5) 错误地配置了网卡导致的启动与关机问题。此时，可以尝试关闭网络适配器并重启计算机，如果问题解决，说明确实是网卡的问题，则需要重新安装网络适配器驱动程序，或者从制造商获取新的驱动程序。

(6) 错误地配置了显卡适配器驱动程序导致的启动与关机问题。此时，可以从其他计算机远程登录，并将当前显卡驱动程序回滚到以前的版本。如果无法做到，就卸载显卡驱动程序之后重新安装。

5.4.1 管理本地系统的重启与关闭

在本地系统上，可以使用如下命令对关机与重启进行管理：

关闭本地系统

```
shutdown /s /t ShutdownDelay /l /f
```

重启本地系统

```
shutdown /r /t ShutdownDelay /l /f
```

取消本地计算机的延迟关机

```
shutdown /a
```

其中，*/T ShutdownDelay*参数用于设置在关机或重启之前的秒数（可选的），*/l*参数用于立即退出当前用户的登录，*/f*参数用于强制关闭运行中的应用程序（而不预先弹出警告信息）。需要说明的是，上面这几个参数都是可选的。下面的实例中，本地系统在60秒的延迟后重启：

```
shutdown /r /t 60
```

最佳实践 在大多数网络环境下，系统正常运行时间是最重要的。系统在重启或关机时都无法为用户提供服务，使得用户无法完成必要的工作，从而导致用户的不快。作为管理员，不要在营业时间关闭或重启系统，而应该在营业时间之外对系统进行关机或重启。如果不得不在营业时间关机，则应该在关机之前通知用户，使得用户可以保存当前的工作，或退出系统，以免造成不必要的损失。

5.4.2 管理远程系统的重启与关闭

对于远程系统，需要使用*/m*参数来指定UNC名或远程系统的IP地址。因此，前面讲述的用于关机、重启、取消延迟的关机等命令的基本语法应该做一些修改，如下面一些实例所示：

关闭远程系统

```
shutdown /s /t ShutdownDelay /l /f /m \\System
```

重启远程系统

```
shutdown /r /t ShutdownDelay /l /f /m \\System
```

取消远程计算机的延迟关机

```
shutdown /a /m \\System
```

下面的实例中，MAILER1在30秒的延迟后重启：

```
shutdown /r /t 30 /m \\Mailer1
```

下面的实例中，IP地址为192.168.1.105的系统将立即重启，正在运行中的应用程序被强迫停止：

```
shutdown /r /f /m \\192.168.1.105
```

5.4.3 添加关机或重启原因与注释

在大多数网络环境下，对关机或重启的原因给出说明是一个好做法。对计划外的关机，你可以在计算机的系统日志中对关机的原因进行记录，这需要对原有命令进行扩展，使其包含如下的参数：

```
/e /c "UnplannedReason" /d MajorCode:MinorCode
```

其中，/e取代了原来的/r开关，/c "UnplannedReason"用于对关机或重启的详细原因进行设置（长度最大为512个字符），/d MajorCode:MinorCode用于设置关机的原因代码。原因代码是任意的，有效的主代码范围为0~255，有效的从代码范围为0~65,535。

对于计划中的重启，参考如下实例：

```
shutdown /r /m \\Mailer1 /c "System Reset" /d 5:15
```

在上面的实例中，重启了MAILER1，并将规划外重启的原因标记为System Reset，原因代码为5:15。

表5-3总结了Windows Vista与Windows Server 2008中关机与重启的常见原因与代码。从表中可以看出，Windows可以生成一些前缀码。比如，E代表Expected（预期中的），U代表Unexpected（预期外的），P代表planned（规划中的），并可以对这些前缀码进行多种组合。

表5-3 关机或重启的常见原因与代码

前 缀 码	主要代码	次要代码	关机或重启类型
U	0	0	其他（计划外的）
E	0	0	其他（计划外的）
EP	0	0	其他（计划内的）
U	0	5	其他失败：系统失去响应
E	1	1	硬件：维护（计划外的）
EP	1	1	硬件：维护（计划外的）
E	1	2	硬件：安装（计划外的）
EP	1	2	硬件：安装（计划外的）
P	2	3	操作系统：升级（计划内的）
E	2	4	操作系统：重新配置（计划外的）
EP	2	4	操作系统：重新配置（计划外的）
P	2	16	操作系统：服务补丁（计划内的）
U	2	17	操作系统：热点补丁（计划外的）
P	2	17	操作系统：热点补丁（计划内的）
U	2	18	操作系统：安全补丁（计划外的）

(续)

前 缀 码	主要代码	次要代码	关机或重启类型
P	2	18	操作系统：服务补丁（计划内的）
E	4	1	应用程序：维护（计划外的）
EP	4	1	应用程序：维护（计划内的）
EP	4	2	应用程序：安装（计划内的）
E	4	5	应用程序：失去响应
E	4	5	应用程序：不稳定
U	5	15	系统失败：终止错误
E	5	19	安全问题
U	5	19	安全问题
EP	5	19	安全问题（计划内的）
E	5	20	失去网络连接（计划外的）
U	6	11	电源故障：Cord Unplugged
U	6	12	电源故障：环境
P	7	0	遗留API关机（计划内的）

对SHUTDOWN命令，只有P:与U:这两个前缀码是可以接受的。比如，对规划中的关机与重启，可以使用前缀码p:来指代一次预期中的关机，如下所示：

```
/c "PlannedReason" /d p:MajorCode:MinorCode
```

下面给出的是另外一个复杂些的实例：

```
shutdown /r /m \\Mailer1 /c "Planned Application Upgrade" /d p:4:2
```

该实例中，重启MAILER1，并将规划中的重启原因记录为Planned Application Upgrade，原因代码为4:2。



到这里为止，本书主要讲解了从命令行管理本地系统与远程系统的工具与技术。本章将讲述如何将事件日志用于系统监控与优化。这里，监控是指经常性地对系统进行检查，以便及时发现存在的问题。优化是指对系统性能进行调试，使其达到最优化的性能指标。

本章将讨论Windows系统中的日志工具，这些工具有助于管理员识别与追踪系统中存在的问题、监控应用程序与服务、维护系统安全等。如果系统出现速度显著降低、行为失常等问题，就需要查看事件日志，以便识别潜在的问题根源。识别了问题根源之后，就可以执行一些维护性或预防性的任务，以便化解或消除这些问题源。通过性能监控，可以观察系统中事件的发生情况，并采取适当的措施对其进行处理。

6.1 Windows 事件日志

在微软Windows系统中，事件是指操作系统中发生的那些显著的、需要用户或管理员加以注意的操作。事件被系统记录在Windows事件日志中，提供了重要的历史信息，有助于完成监控系统、维护系统安全、解决问题、进行系统诊断等任务。定期地检查事件日志不仅是重要的，也是基础性的工作。管理员应该密切监控每台商业服务器的事件日志，同时也要确保工作站进行了正确的配置，以便对重要的系统事件进行追踪。对服务器，要确保系统是安全的，应用程序与服务是正常运行的，并且服务器不能出现影响系统性能的错误。对工作站，要确保那些有助于系统维护和解决问题的事件被记录下来，并且能在需要的时候方便地访问事件日志。

用于管理事件日志的Windows服务称为Windows Event Log服务。该服务启动后，Windows会记录重要的事件信息。系统中有哪些可用的事件日志依赖于系统本身的角色以及安装了哪些服务。两种通用的日志文件类型包括下面几个。

- **Windows日志**。操作系统使用的用于记录通用系统事件（与应用程序、安全性、启动、系统组件等相关）的事件日志。
- **应用程序与服务日志**。特定的应用程序与服务使用的、用于记录应用程序或服务特定事件的事件日志。

具体来讲，有如下的一些事件日志。

- **应用程序日志**。该日志记录了与特定应用程序相关的重要事件。比如，Exchange Server会记录与邮件交换相关的事件，包括信息存储、邮箱、服务声明等事件。默认情况下，应用程序日志存储在%SystemRoot\System32\Winevt\Logs\Application.Evtx。
- **目录服务日志**。在域控制器上，该日志记录了来自活动目录域服务（AD DS）的事件，包括

目录启动、全局编目以及完整性检查等。默认情况下，目录服务日志存储在%SystemRoot%\System32\Winevt\Logs\Directory Service.Evtx。

- DNS服务器日志。在DNS服务器上，该日志记录了DNS查询、响应以及其他相关的DNS活动。默认情况下，DNS服务器日志存储在%SystemRoot%\System32\Winevt\Logs\DNS Server.Evtx。
- DFS复制日志。在使用DFS复制的域控制器上，该日志记录了系统中的文件复制活动，包括服务状态与控制、系统卷中数据扫描、复制集管理等事件。默认情况下，DFS复制日志存储在%SystemRoot%\System32\Winevt\Logs\DFS Replication.Evtx。
- 文件复制服务日志。该日志记录了系统中的文件复制活动。默认情况下，该日志存储在%SystemRoot%\System32\Winevt\Logs\File Replication Service.Evtx。
- 转发事件日志。系统中配置了事件转发功能时，该日志会记录从其他服务器转发而来的事件。转发事件日志默认的存储位置为%SystemRoot%\System32\Winevt\Logs\Forwarded-Events.Evtx。
- 硬件事件日志。系统中配置了硬件子系统事件报告机制时，该日志会记录报告给操作系统的硬件事件。硬件事件日志默认的存储位置为%SystemRoot%\System32\Winevt\Logs\HardwareEvents.Evtx。
- Microsoft\Windows。用于追踪特定的Windows服务与功能相关事件的一组日志，以组件类型与事件类别进行组织。
- 安全性日志。该日志记录了与安全性相关的事件，比如登录/注销、特权使用、资源访问等。默认情况下，安全性日志存储在%SystemRoot%\System32\Winevt\Logs\Security.Evtx。

注解 要访问安全性日志，必须具备Manage Auditing And Security Log这一用户权限。默认情况下，管理员组中的成员具备这一权限。在*Windows Server 2008 Administrator's Pocket Consultant* (Microsoft Press,2008)一书的第10章中，讲述了用户权限的分配问题。

- 安装日志。该日志记录了操作系统或其组件在安装时的相关事件。安装日志的默认存储位置为%SystemRoot%\System32\Winevt\Logs\Setup.Evtx。
- 系统日志。该日志记录了来自操作系统或其组件的事件，比如某服务启动失败、驱动程序初始化、系统范围的消息、其他与系统相关的消息等。默认情况下，系统日志存储在%SystemRoot%\System32\Winevt\Logs\System.Evtx。
- Windows PowerShell日志。该日志记录了与使用Windows PowerShell相关的事件。Windows PowerShell日志默认的存储位置为%SystemRoot%\System32\Winevt\Logs\Windows PowerShell.Evtx。

在事件日志中，每一事件都带有严重程度，从信息性的事件到一般的警告事件、到严重事件（比如关键性错误与失败等）。事件所属类别是由其事件级别标示的，事件级别包括下面5个。

- 信息。表示系统中发生了信息性的事件，通常与某一成功的动作相关。
- 警告。表示一般性的警告，一般用于提醒用户防止以后的系统问题。
- 错误。表示系统中发生了关键性的错误，比如某服务启动失败。
- 成功审核。表示通过审核进行追踪的某个操作成功执行，比如特权使用。
- 失败审核。表示通过审核进行追踪的某个操作执行失败，比如登录失败。

注解 在众多的事件类型中，应该密切关注的是警告事件与错误事件。在产生这两类事件而又不能确定其原因后，你应该进一步研究，以便决定是否有必要采取进一步的动作。

除事件级别外，每一事件还有如下一些常见的属性。

- **日期与时间**。记录了事件发生的日期与时间。
- **事件源**。记录了事件源，比如应用程序、服务或系统组件等。事件源有助于找到导致事件的根源。
- **事件ID**。使用数字形式的标识符来记录特定的事件。事件ID是由事件源生成的，用于唯一性地标记事件。
- **事件类别**。表示事件所属的类别，有时可用于进一步地描述相关操作。每一事件源都有自己的事件类别。比如，对于安全性事件源，其事件类别包括登录/注销、特权使用、策略改变、账号管理等。
- **用户**。表示导致生成该事件的用户账号。用户可以包括特殊的标识符，比如本地服务、网络服务、匿名登录，以及实际的用户名。用户账号也可以标记为不适用，表示该场景下不适用用户账号。
- **计算机**。表示导致发生该事件的计算机。
- **描述**。为事件提供了详尽的描述信息，也可以包括关于从哪里找到更多信息（以便解决或处理问题）等内容。在事件查看器中，双击某日志条目，就可以查看描述字段。
- **数据**。事件的相关数据或错误代码等输出信息。

可用于事件管理的图形界面工具是事件查看器。在命令行中，键入`eventvwr`命令可以启动本地计算机上的事件查看器，而如果要启动远程计算机上的事件查看器，则需要键入`eventvwr ComputerName`命令，其中`ComputerName`是远程计算机名。与大多数GUI工具类似，事件查看器易于使用，并且对于某些特定的管理任务，只能使用该工具。比如，如果需要控制事件日志的大小、指定事件日志的处理方式、对事件日志进行存档等，就必须使用事件查看器，而无法在命令行完成这些任务。

Windows Vista与Windows Server 2008提供了一些不同的工具与技术，可用于在命令行对事件日志进行处理，主要包括下面3个。

- **Powershell Get-Eventlog**。搜索事件日志并收集那些匹配特定标准的事件条目。在脚本中，可以使用Powershell `Get-Eventlog`来检查多个日志中的事件，并将结果存储到文件中，以便于追踪信息、警告与错误等不同严重程度的事件。
- **Eventcreate**。在事件日志中创建自定义事件。根据计划（或者作为例行维护任务的一部分）运行自定义脚本时，你可能希望在事件日志中记录这些动作，`Eventcreate`可以用于完成这一任务。
- **自定义视图**。使用XPath查询来创建自定义或过滤后的事件日志视图，以便快速、方便地发现匹配特定标准的事件。由于XPath查询可以在兼容的系统上重用，因此，在目标计算机上重新运行该查询，就可以重建自定义或过滤后的事件日志视图。

真实场景 监控系统事件并不是一件根据个人兴趣随意为之的事情。相反，这是一项应该经常地、彻底地执行的任务。对于服务器，应该至少每天检查一次事件日志。对于工作站，应该在需要的时候检查其上的事件日志，比如在用户声称存在问题的时候。

6.2 查看与过滤事件日志

通过Windows PowerShell的Get-Eventlog cmdlet^①，可以从事件日志中获取详尽的信息。使用Get-Eventlog时，不要忽略自动化的威力。你没有必要每次从Windows PowerShell提示符中手工运行该命令，而是可以创建一个脚本来自动化地查询事件日志，并将结果保存到文件中。如果将结果文件复制到内部网服务器上的发布文件夹，还可以使用Web浏览器来对事件列表进行查看。通过这种做法，不仅可以节省时间，还为事件日志检查提供了一个单独的场所，以便确定是否存在需要进一步研究和解决的问题。

注解 本书讨论的工具都是从命令行中完成相应任务的最佳工具。在这一场景中，通过命令行从事件日志中提取信息的首选工具是Windows PowerShell。遗憾的是，尽管本书介绍了Windows PowerShell并讨论了cmdlet，但囿于篇幅和定位，本书没有提供关于Windows PowerShell更多的信息。要获取Windows PowerShell相关的更进一步的信息，推荐参阅在*Windows PowerShell Administrator's Pocket Consultant* (Microsoft Press, 2008) 一书。

6.2.1 查看事件

Get-Eventlog需要在Windows PowerShell提示符中运行，其基本语法格式为：

```
get-eventlog "LogName"
```

其中，LogName为待处理的事件日志名，比如“应用程序”、“系统”、“目录服务”等。下面的实例检查应用程序日志：

```
get-eventlog "Application"
```

注解 理论上，上面命令行实例中的引号只有在日志名中包含空格时才是必要的，比如DNS Server、Directory Service、File Replication Service等日志名。不过，我建议任何时候都使用引号将日志名包含起来，养成这一习惯后，就不会在需要引号的时候忘记使用，从而避免脚本或计划任务的失败。

执行上述命令后，其输出类似于如下的格式：

Index	Time	Type	Source	EventID	Message
15959	Mar 20 16:56	Error	MSExchange System...	4001	A transient failure
			has occurred. The problem may resolve its...		
15958	Mar 20 16:55	Error	MSExchange System...	4001	A transient failure
			has occurred. The problem may resolve its...		
15957	Mar 20 16:54	Error	MSExchange System...	4001	A transient failure
			has occurred. The problem may resolve its...		
15956	Mar 20 16:53	Error	MSExchange System...	4001	A transient failure
			has occurred. The problem may resolve its...		

可以看出，输出信息展示了索引、时间、类型、事件源、事件ID、事件消息属性等内容。索引是

① 置的Windows PowerShell命令称为cmdlet，比如Get-Eventlog。——译者注

该事件在事件日志中所处的位置，该实例中为15,956~15,959。此外，在Get-Eventlog后跟随日志名后，-Logname参数是暗含的，但也可以直接指定-Logname参数，如下所示：

```
get-eventlog -logname"security"
```

默认情况下，Get-Eventlog会返回指定的事件日志中的每一个事件（从最近事件到最早事件）。显然，大多数情况下，返回的信息量会大到无法处理，因而需要对事件进行过滤，以便使事件量处于可处理的范围。一种简单的事件过滤方法是在命令行中指定只希望查看最新事件。比如，你可能只需要查看事件日志中最新的100条事件。

通过使用-Newest参数，就可以将返回的事件限制在最新事件的范围。如下的实例列出了安全事件日志中最新的100条事件：

```
get-eventlog"security"-newest 50
```

要注意的是，与前面讲过的一些命令行工具不同的是，Get-Eventlog是一个Windows PowerShell cmdlet。如果是第一次使用Windows PowerShell，要确保该功能已经安装在系统中。如果不希望调用一个单独的Windows PowerShell实例，就可以在调用Windows PowerShell时使其只运行Get-Eventlog cmdlet，如下所示：

```
powershell.exeget-eventlog -logname"security"
```

你也可以将这一命令插入到脚本中。在批处理脚本中，这一命令将调用Windows PowerShell，执行Get-Eventlog cmdlet，之后退出Windows PowerShell。

6.2.2 过滤事件

使用Get-Eventlog的一个原因是，该工具可以在结果集中对事件进行分组与过滤。通过将事件按类型分组，可以更方便地将信息事件与关键事件、警告事件与错误事件等不同类型的事件区分开来。通过按事件源进行分组，可以更方便地对来自特定源的事件进行追踪。通过按事件ID进行分组，可以更方便地对特定事件的重现进行关联分析。

通过下面讲述的方法和技术，可以根据事件源、事件ID、事件条目类型、生成时间等对事件进行分组。

- (1) 获取需要处理的事件，并将其存储在\$e变量中：

```
$e = get-eventlog -newest 100 -logname "application"
```

- (2) 根据特定的属性，使用Group-Object cmdlet对存储在\$e变量中的事件进行分组。本例中是根据事件ID进行分组：

```
$e | group-object-property eventid
```

另一种处理事件的方法是根据特定属性进行排序。通过下面讲述的方法和技术，可以根据事件源、事件ID、事件条目类型、生成时间等对事件进行排序。

- (1) 获取需要处理的事件，并将其存储在\$e变量中：

```
$e = get-eventlog -newest 100 -logname "application"
```

- (2) 根据特定的属性，使用Sort-Object cmdlet对存储在\$e变量中的事件进行排序。本例中是根据事件条目类型进行排序：

```
$e | sort-object-property entrytype
```


典型情况下，并不需要查看系统中生成的每一事件。更多的时候，只需关注警告事件与关键性错误事件，这也是事件过滤的主要用途所在。通过事件过滤器，可以只包含那些匹配特定标准的事件。下面给出一个实例，通过搜索事件条目类型属性来寻找关键字`error`，来过滤出错误事件。

(1) 获取需要处理的事件，并将其存储在`$e`变量中：

```
$e = get-eventlog -newest 500 -logname "application"
```

(2) 对存储在`$e`变量中的事件对象的指定属性，使用`Where-Object` cmdlet搜索匹配的文本。本例中搜索事件条目类型为`error`的事件：

```
$e | where-object {$_.EntryType -match "error"}
```

`Where-Object` cmdlet使用的搜索算法不区分大小写，也就是说，输入`Error`、`error`或`ERROR`等都可以匹配错误事件。此外，`Where-Object`采用的是部分匹配策略。因此，如果要搜索警告、关键性事件、信息事件等事件时，不需要输入完整的事件条目类型，而只需分别输入`warn`、`crit`或`info`，如下所示：

```
$e = get-eventlog -newest 100 -logname "application"
$e | where-object {$_.EntryType -match "warn"}
```

也可以使用`Where-Object`处理其他的事件对象属性。比如，下面的实例搜索事件源中包含文本`MSDTC`的事件：

```
$e = get-eventlog -newest 500 -logname "application"
$e | where-object {$_.Source -match "MSDTC"}
```

下面的实例搜索事件ID 15001：

```
$e = get-eventlog -newest 500 -logname "application"
$e | where-object {$_.EventID -match "15001"}
```

如果希望将这些工作自动化，你可以创建一个Windows PowerShell脚本，获取需要查看的事件信息，并将其写入到文本文件。参考如下实例：

```
$e = get-eventlog -newest 100 -logname "system"
$e | where-object {$_.EntryType -match "error"} > currentlog.txt
```

```
$e = get-eventlog -newest 500 -logname "application"
$e | where-object {$_.EntryType -match "error"} >> currentlog.txt
```

```
$e = get-eventlog -newest 500 -logname "directory service"
$e | where-object {$_.EntryType -match "error"} >> currentlog.txt
```

上面的脚本命令中，检查了系统、应用程序、目录服务等事件日志，并将任意输出结果写入到`CorpIntranet01`上的网络共享位置。如果某事件日志中的最新500条事件中包含了错误事件，则错误事件被写入到`Currentlog.txt`文件中。由于第一个重定向字符是重写(`>`)，而余下的入口使用的重定向字符是添加(`>>`)。因此，该脚本每次运行时，现有的`Currentlog.txt`文件都将被重写，从而确保事件的最新。要更进一步地自动化，可以创建一个计划任务，使其每天运行该脚本，或在特定的时间间隔后运行该脚本。

Windows PowerShell脚本文件的文件扩展名为`.ps1`（注意，依次为字母P、字母S、数字1），要在Windows PowerShell提示符中运行Windows PowerShell脚本，需要键入脚本名，后面跟随扩展名（可选的）。在脚本名前，必须指定脚本的全路径，即便脚本在当前目录。要指明当前目录，可以使用目录名，也可以使用句点(`.`)来代表当前目录。根据这些约定，如果将Windows PowerShell脚本文件以文

件名CheckEvents.ps1存储在当前目录，要运行该文件，就需要在Windows PowerShell提示符中输入.\checkEvents.ps1。

6.3 向事件日志中写入自定义事件

运行自动化脚本、计划任务以及自定义应用程序时，你可能希望这些自动化脚本、计划任务以及自定义应用程序向事件日志中写入自定义事件。比如，如果脚本正常运行，就向应用程序日志中写入信息事件，以便确认脚本正常执行完毕。同样地，如果脚本不能正常运行并产生错误，就向应用程序日志中写入错误事件或警告信息，通过查看应用程序事件日志，就可以获知脚本运行出错，需要进一步检查其出错原因。

提示 使用`%ErrorLevel%`可以追踪脚本中发生的错误。这一环境变量可以追踪最近使用命令的退出代码。如果命令正常执行，则错误级别为0；如果命令执行出错，则错误级别设置为非0值。要了解关于错误级别的更多知识，可以参考第3章3.4节。

要创建自定义事件，可以使用Eventcreat工具。自定义事件可以写入到除安全日志以外的任意可用的事件日志中，内容上可以包含事件源、事件ID以及想要添加的描述信息。Eventcreat的语法格式如下：

```
eventcreate /l LogName /so EventSource /t EventType /id EventID  
/d EventDescr
```

其中各字段含义如下。

- **LogName**。设置事件将要写入的事件日志名。如果事件日志名中包含空格，则应该使用引号将其封装，比如DNS Server。

提示 要注意不能向安全事件日志中写入自定义事件，其他事件日志中都可以写入。写入时，首先使用待追踪的事件源写入一个dummy事件，与该事件源相关的初始化事件会写入到应用程序日志，之后可以使用该事件源与指定的日志操作自己的自定义事件。

- **EventSource**。指定要使用的事件源。EventSource可以是任意的字符串，如果字符串中包含空格，则应该使用引号将其封装，比如Event Tracker。大多数情况下，使用事件源是为了识别应用程序、任务或产生错误事件的脚本。

警告 在使用事件源将事件写入到事件日志之前（这里，事件源是自己命名的），要对事件源的命名进行认真规划。每一个事件源名必须是独一无二的，不能与系统中安装的应用程序或服务所使用的事件源重名，也不能使用由Windows角色、角色服务与功能使用的事件源名。比如，不能使用DNS、W32Time、Ntfrs等作为自定义事件的事件源名，因为Windows Server 2008已经使用了这些名。此外，一旦为某特定事件日志使用了某个事件源，则该事件源就与该日志绑定在一起。比如，不能将EventChecker同时作为MAILER1上应用程序日志与系统日志的事件源。如果先前已经使用EventChecker作为事件源将事件写入到应用程序日志，后面又试图使用EventChecker作为事件源将事件写入到系统日志，系统会给出错误信息：“错误：源存在于Application日志中。源不能被复制”。

- **EventType**。将事件类型设置为信息、警告、错误等类型，但不能将其设置为“成功审核”或“失败审核”等类型。因为这两种类型是与安全日志相关的，而自定义事件不允许写入到安全日志。
- **EventID**。设置事件的事件ID，其取值范围从1~1,000。在设置事件ID之前，最好对常见事件进行分类，之后为每一类事件分别设定一个事件ID取值范围。比如，事件ID在100以内的为通常事件，事件ID取值范围100~200的为状态事件，事件ID取值范围200~500的为警告事件，事件ID取值范围500~900的则为错误事件。取值范围的划分是任意的，只要能有效区分事件类型即可。
- **EventDescr**。设置事件的相关描述信息，可以是任意的字符串，注意要包含在引号中。

注解 默认情况下，在本地系统上，Eventcreate以当前登录用户的权限运行。必要的时候，如果需要查询远程计算机上的任务并指定运行权限，可以使用类似于/S Computer /u [Domain\]User[/P Password]的语法格式。其中，Computer为远程计算机名或IP地址，Domain为可选的域名（用户账号存在于该域内），User为要用户账号名（要使用的就是该用户账号的权限），Password为该用户账号的口令（可选的）。

要进一步了解如何使用Eventcreate，参考如下示例。

在应用程序日志中创建信息事件，事件源为Event Tracker，事件ID为209：

```
eventcreate /l "application" /t information /so "Event Tracker"  
/id 209 /d "evs.bat script ran without errors."
```

在系统日志中创建警告事件，事件源为CustApp，事件ID为511：

```
eventcreate /l "system" /t warning /so "CustApp" /id 511 /d  
"sysck.exe didn't complete successfully."
```

在MAILER1的系统日志中创建错误事件，事件源为SysMon，事件ID为918：

```
eventcreate /s Mailer1 /l "system" /t error /so "SysMon" /id 918  
/d "sysmon.exe was unable to verify write operation."
```

6.4 创建与使用保存的查询

在设计Windows Vista与Windows Server 2008时，微软对事件查看器的过滤与查询功能做了很大的增强。受益于这种增强，事件查看器可以支持XPath查询，以便创建自定义视图和过滤事件日志。XPath是一种non-XML语言，用于识别XML文档中的特定部分。事件查看器可以使用XPath表达式在源日志中匹配与选择特定的部分，并将其复制到目标日志，以便创建自定义或过滤的视图。

在事件查看器中创建自定义或过滤的视图时，可以将XPath查询复制并保存到事件查看器的自定义视图文件中。以后再次运行这一查询时，就可以在任意运行Windows Vista与Windows Server 2008的计算机上重建自定义或过滤的视图。比如，你可以为应用程序日志创建一个过滤的视图，来识别Microsoft SQL Server中存在的问题。在创建的同时，可以将相关的XPath查询保存到自定义视图中，通过这一文件，就可以很便利地在组织内其他计算机上为应用程序日志创建过滤的视图。

事件查看器会自动地为事件日志创建几种过滤的视图，这些过滤的视图在自定义视图节点下列出。选择管理员事件节点时，会看到事件日志中所有错误事件与警告事件。扩展服务器角色节点并选择其中一个特定角色的视图时，可以看到该角色相关的所有事件。

通过如下步骤，可以创建并保存自己的自定义视图。

(1) 启动事件查看器，方法是依次单击“开始”、“管理工具”、“事件查看器”。

(2) 选择自定义视图节点，在操作菜单中，单击“创建自定义视图”。

(3) 在图6-1所示的“创建自定义视图”对话框中，使用Logged列表选择事件记录的时间帧。你可以选择包含事件的开始时间，可以是任何时间、前一个小时、过去的12小时、过去的24小时、最后7天、最后30天等。

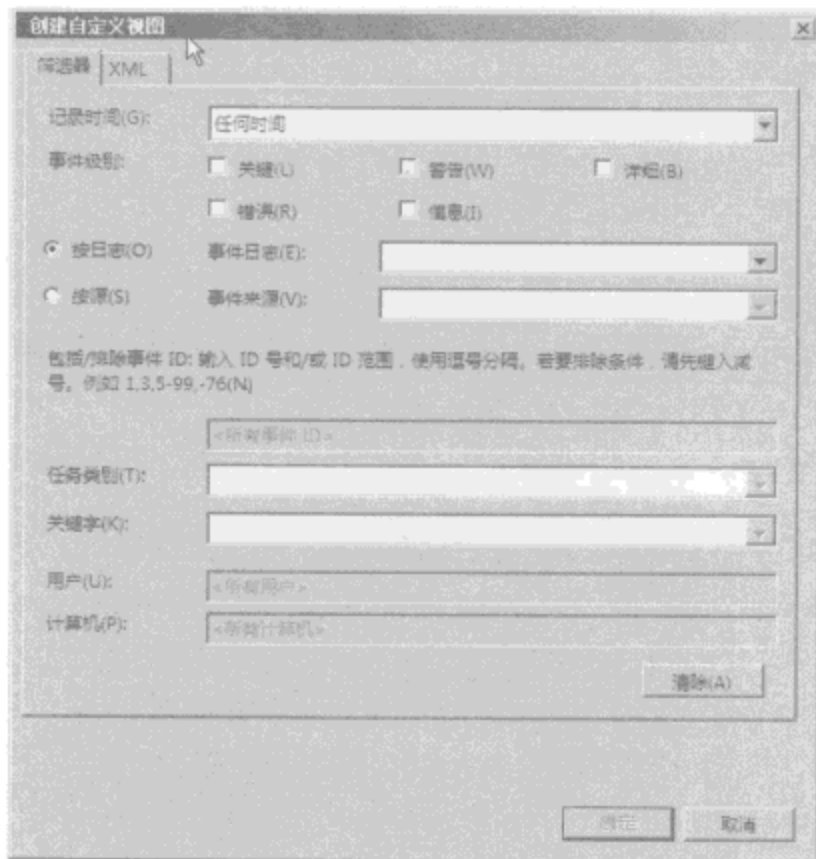


图6-1 创建筛选器，以便指定显示事件的类型

(4) 使用事件级别复选框指定要包含的事件级别，选择详细可以获取更多的详细资料。

(5) 为特定的一组日志或事件源创建自定义视图。

□ 使用事件日志列表选择要包含的事件日志。可以使用相应的复选框选择多个事件日志，如果选择了特定的事件日志，则其他日志会被排除在外。

□ 使用事件来源列表选择要包含的事件源。可以使用相应的复选框选择多个事件源，如果选择了特定的事件源，则其他事件源会被排除在外。

(6) 使用用户与计算机对话框来指定应该包含的用户与计算机（可选的）。如果不指定这一选项，则所有用户与计算机生成的事件都会包含在其中。

(7) 单击XML选项卡来显示相关的XPath查询，如图6-2所示。

(8) 单击“确定”，关闭“创建自定义视图”对话框。在图6-3中所示的“将筛选器保存到自定义视图”对话框中，为该自定义视图键入名称及描述信息。

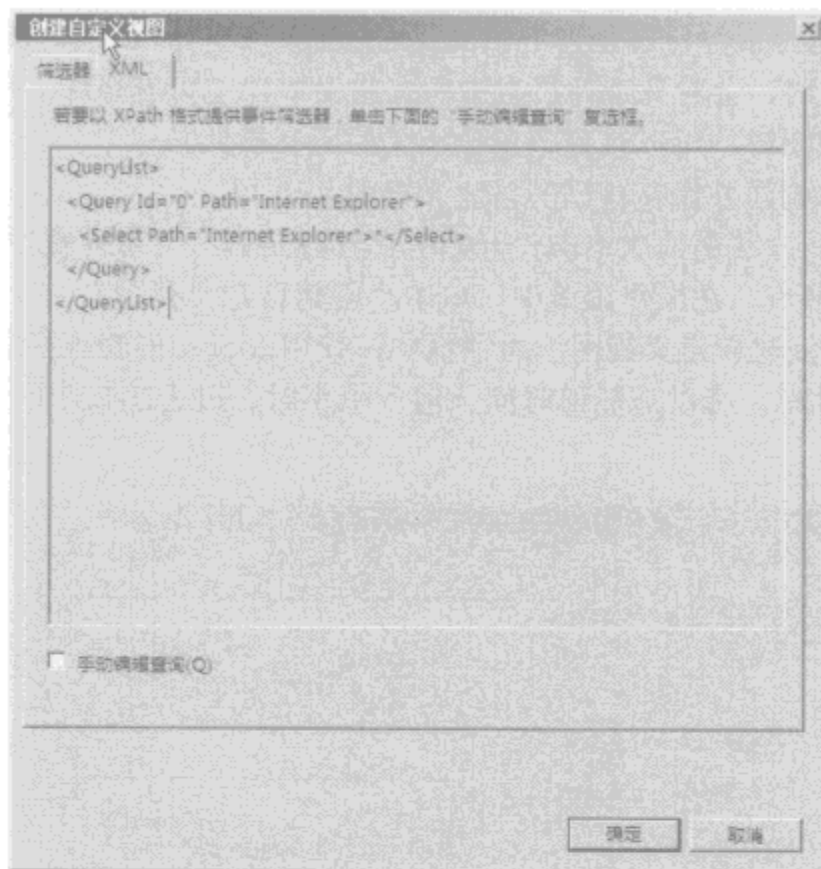


图6-2 查看相关的XPath查询

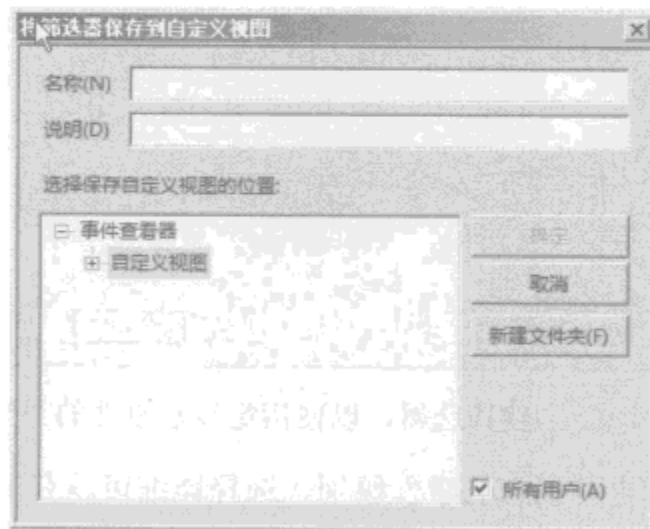


图6-3 保存筛选的视图

(9) 选择自定义视图的保存位置。默认情况下，自定义视图保存在自定义视图节点，但也可以为其创建一个新节点——单击“新文件夹”，输入新文件夹名，之后单击“确定”即可。

(10) 单击“确定”，关闭“将筛选器保存到自定义视图”对话框。至此，应该可以看到过滤后的事件列表。

(11) 鼠标右击“自定义视图”，选择“导出自定义视图”，使用另存为对话框选择保存位置，为事件查看器自定义视图输入文件名。

自定义视图中包含了此前已在XML选项卡中显示的XPath查询。事件日志阅读组的成员、管理员组成员以及其他具有适当权限的用户，可以运行该查询，以便查看远程计算机上的事件，如下所示：


```
eventvwr ComputerName /v: QueryFile
```

其中，*ComputerName*是远程计算机名（要检查其上的事件），*QueryFile*是包含XPath查询的自定义视图的文件名或全路径，比如：

```
eventvwr mailserver25 /v: importantevents.xml
```

启动事件查看器时，可以在自定义视图节点下发现自定义视图。

6.5 性能监控：基础

前面讲解了如何查看、筛选与创建事件，本节将讲解可用于计算机性能监控的相关技术。Windows Vista与Windows Server 2008包含了几款可用于性能监控的工具，本节将讲述命令行工具Typeperf，第7章将讲述可用于性能监控的其他工具。

6.5.1 理解如何在命令行中进行性能监控

Typeperf是一款用于实时追踪并显示性能信息的命令行工具，该工具可以根据已配置的性能监控参数收集信息，并将其在命令行中显示出来。待监控的性能项目是由如下3个部分来定义的。

- **性能对象**。代表任意具备一组可测量属性的系统组件。性能对象可以是操作系统的物理组成部分，比如内存、处理器或页面文件。也可以是逻辑组件，比如逻辑磁盘或打印队列。还可以是软件组件，比如进程或线程。
- **性能对象实例**。代表性能对象的单一出现。如果特定性能对象有多个实例，比如，计算机有多个处理器或多个磁盘驱动器，就可以使用性能对象实例来指定该对象的某个单一出现。你也可以选择追踪与监控某性能对象的所有实例，比如监控系统中所有处理器。
- **性能计数器**。代表性能对象中可测量的属性。比如，对页面文件，可以使用%Usage计数器来度量其使用的百分比。

在Windows Vista与Windows Server 2008的标准安装中，有很多可以监控的性能对象。添加服务、应用程序与组件后，又会增加一些新的性能对象。比如，在服务器上安装DNS后，该服务器上就增加了一个可供监控的DNS对象。

表6-1中总结了实际工作中需要监控的常见性能对象，其中列出的每一个性能对象都有一组可以追踪与监控的计数器。通常，管理员监控最多的性能对象是内存、物理磁盘以及处理器。

表6-1 最常追踪的性能对象

性能对象	描 述
缓存	监控文件系统缓存，缓存是一块反映应用程序I/O活动的内存区域
数据库—>实例	监控操作系统使用的嵌入式数据库管理系统实例的性能
IPv4	监控IPv4通信及相关活动
IPv6	监控IPv6通信及相关活动
逻辑磁盘	监控计算机上的逻辑卷
内存	监控系统缓存（包括换页池内存以及非换页池内存）、物理内存、虚拟内存的性能
网络接口	监控计算机上配置的网络适配器
对象	监控计算机上事件、互斥对象、进程、段、信号以及线程的数量

(续)

性能对象	描 述
页面文件	监控页面文件的当前使用与峰值使用情况
物理磁盘	监控硬盘读写活动, 以及数据传输、硬件失效与软件失效等情况
打印队列	监控打印任务、打印池以及打印队列的活动
进程	监控计算机上运行的所有进程
处理器	监控处理器的空闲时间、空闲状态、使用情况、延迟的过程调用以及中断等情况
服务器	监控计算机与网络的通信情况, 以及一些重要的统计信息, 包括登录错误、访问错误以及用户会话等
服务器工作队列	监控线程与客户端请求
系统	监控系统级的计数器, 包括进程、线程、线程上下文切换、文件系统控制操作、系统调用以及系统正常运行时间等
TCPv4	监控TCPv4通信及相关活动
TCPv6	监控TCPv6通信及相关活动
线程	监控系统中运行的所有线程, 也可以根据进程ID来检查单独的线程的使用情况统计信息
UDPv4	监控UDPv4通信及相关活动
UDPv6	监控UDPv6通信及相关活动

6.5.2 追踪性能数据

通过Typeperf, 可以将性能数据写入到命令行或日志文件。使用Typeperf时, 关键是指定待追踪的性能计数器的路径名, 性能计数器路径具有如下的语法格式:

```
\\ComputerName\ObjectName\ObjectCounter
```

其中, *ComputerName*为本地或远程计算机的计算机名或IP地址, *ObjectName*为性能对象名, *ObjectCounter*为性能对象计数器名。比如, 如果需要追踪FileServer42上可用内存情况, 可以使用如下命令:

```
typeperf "\\fileserver42\memory\available mbytes"
```

注解 本例中, 由于计数器路径中包含空格, 因此需要使用双引号对其进行封装。此外, 尽管不是所有情况都必须使用双引号, 但总是使用双引号进行封装是一个好做法。

通过将星号(*)作为计数器名, 可以很方便地对某性能对象的所有计数器进行追踪, 如下所示:

```
typeperf "\\fileserver42\Memory\*"
```

该实例中, 实现了对内存对象所有计数器的追踪。

如果某性能对象有多个实例, 比如处理器或逻辑磁盘对象, 就必须指定具体的对象实例。其基本语法格式为:

```
\\ComputerName\ObjectName(ObjectInstance)\ObjectCounter
```

其中, 具体的对象实例使用花括号进行封装, 跟随在对象名之后。如果某对象有多个实例, 通过将_Total作为实例名, 可以实现对该对象所有实例的处理。通过某实例的标识符, 可以处理性能对象的某个具体实例。比如, 如果想要检查Processor\%Processor Time计数器, 可以使用如下命令处理所有

处理器实例:

```
typeperf "\\fileserver42\Processor(_Total)\%Processor Time"
```

下面的命令则用于处理一个特定的处理器实例:

```
typeperf "\\fileserver42\Processor(0)\%Processor Time"
```

其中, Processor(0)指代系统中的第一个处理器。

Typeperf有很多可用的参数,表6-2对其进行了总结。

默认情况下, Typeperf将其输出信息以逗号分隔的列表形式写入到命令行。你也可以使用-O参数对输出信息进行重定向,使用-F参数对输出格式进行设置。设置输出格式时,CSV代表逗号分隔的文本文件,TSV代表制表符分隔的文本文件,BIN代表二进制文件,SQL代表SQL二进制文件。参考如下实例:

```
typeperf "\\fileserver42\Memory\*" -o memperf.bin -f bin
```

表6-2 Typeperf的参数

参 数	描 述
-cf <filename>	识别包含了待监控性能计数器列表的文件
-config <filename>	识别包含了命令选项的设置文件
-f <CSV TSV BIN SQL>	设置输出文件格式。默认情况下,对以逗号分隔开的值,输出文件为.csv文件
-o <filename>	设置输出文件或SQL数据库的路径
-q [object]	列出某指定对象已安装的计数器
-qx [object]	列出已安装计数器与实例
-s <ComputerName>	设置待监控的远程计算机(如果计数器路径中没有指定计算机)
-sc <samples>	设置要收集的采样数
-si <[hh:]mm:ss>	设置采样之间的时间间隔,默认为1秒
-y	对所有问题,设置回答信息为“是”,从而不再弹出提示符

上面的命令中,对内存对象的所有计数器进行了追踪,并将输出信息写入到当前目录下名为MemPerf.bin的二进制文件中。

如果需要确定某性能对象所有可用的计数器,可以键入typeperf-q命令,其后跟随对象名。比如,如果输入如下命令:

```
typeperf -q Memory
```

就会得到一个类似于如下的可用计数器列表:

```
\memory\Page Faults/sec
\memory\Available Bytes
\memory\Committed Bytes
\memory\Commit Limit
\memory\Write Copies/sec
\memory\Transition Faults/sec
\memory\Cache Faults/sec
\memory\Demand Zero Faults/sec.
\memory\Pages/sec
\memory\Pages Input/sec
\memory\Page Reads/sec
```

```

\memory\Pages Output/sec
\memory\Pool Paged Bytes
\memory\Pool Nonpaged Bytes
\memory\Page Writes/sec
\memory\Pool Paged Allocs
\memory\Pool Nonpaged Allocs
\memory\Free System Page Table Entries
\memory\Cache Bytes
\memory\Cache Bytes Peak
\memory\Pool Paged Resident Bytes
\memory\System Code Total Bytes
\memory\System Code Resident Bytes
\memory\System Driver Total Bytes
\memory\System Driver Resident Bytes
\memory\System Cache Resident Bytes
\memory\% Committed Bytes In Use
\memory\Available KBytes
\memory\Available MBytes
\memory\Transition Pages RePurposed/sec
\memory\Free & Zero Page List Bytes
\memory\Modified Page List Bytes
\memory\Standby Cache Reserve Bytes
\memory\Standby Cache Normal Priority Bytes
\memory\Standby Cache Core Bytes

```

如果某对象有多个实例，通过-Qx参数，可以列出与某具体实例对应的已安装的计数器。如下所示：

```
typeperf -qx PhysicalDisk
```

执行后，输出信息中给出了一个很长的可用计数器列表，并分别与各自的对象实例对应：

```

\PhysicalDisk(0 E: C:)\Current Disk Queue Length
\PhysicalDisk(1 D:)\Current Disk Queue Length
\PhysicalDisk(2 I:)\Current Disk Queue Length
\PhysicalDisk(3 J:)\Current Disk Queue Length
\PhysicalDisk(4 K:)\Current Disk Queue Length
\PhysicalDisk(5 L:)\Current Disk Queue Length
\PhysicalDisk(6 N:)\Current Disk Queue Length
\PhysicalDisk(7 O:)\Current Disk Queue Length
\PhysicalDisk(8 P:)\Current Disk Queue Length
\PhysicalDisk(9 Q:)\Current Disk Queue Length
\PhysicalDisk(_Total)\Current Disk Queue Length
\PhysicalDisk(0 E: C:)\% Disk Time
\PhysicalDisk(1 D:)\% Disk Time
\PhysicalDisk(2 I:)\% Disk Time
\PhysicalDisk(3 J:)\% Disk Time
\PhysicalDisk(4 K:)\% Disk Time
\PhysicalDisk(5 L:)\% Disk Time
\PhysicalDisk(6 N:)\% Disk Time
\PhysicalDisk(7 O:)\% Disk Time
\PhysicalDisk(8 P:)\% Disk Time
\PhysicalDisk(9 Q:)\% Disk Time
\PhysicalDisk(_Total)\% Disk Time
\PhysicalDisk(0 E: C:)\Avg. Disk Queue Length
\PhysicalDisk(1 D:)\Avg. Disk Queue Length

```



```

\PhysicalDisk(2 I:)\Avg.Disk Queue Length
\PhysicalDisk(3 J:)\Avg.Disk Queue Length
\PhysicalDisk(4 K:)\Avg.Disk Queue Length
\PhysicalDisk(5 L:)\Avg.Disk Queue Length
\PhysicalDisk(6 N:)\Avg.Disk Queue Length
\PhysicalDisk(7 O:)\Avg.Disk Queue Length
\PhysicalDisk(8 P:)\Avg.Disk Queue Length
\PhysicalDisk(9 Q:)\Avg.Disk Queue Length
...
\PhysicalDisk(0 E: C:)\% Idle Time
\PhysicalDisk(1 D:)\% Idle Time
\PhysicalDisk(2 I:)\% Idle Time
\PhysicalDisk(3 J:)\% Idle Time
\PhysicalDisk(4 K:)\% Idle Time
\PhysicalDisk(5 L:)\% Idle Time
\PhysicalDisk(6 N:)\% Idle Time
\PhysicalDisk(7 O:)\% Idle Time
\PhysicalDisk(8 P:)\% Idle Time
\PhysicalDisk(9 Q:)\% Idle Time
\PhysicalDisk(_Total)\% Idle Time
\PhysicalDisk(0 E: C:)\Split IO/Sec
\PhysicalDisk(1 D:)\Split IO/Sec
\PhysicalDisk(2 I:)\Split IO/Sec
\PhysicalDisk(3 J:)\Split IO/Sec
\PhysicalDisk(4 K:)\Split IO/Sec
\PhysicalDisk(5 L:)\Split IO/Sec
\PhysicalDisk(6 N:)\Split IO/Sec
\PhysicalDisk(7 O:)\Split IO/Sec
\PhysicalDisk(8 P:)\Split IO/Sec
\PhysicalDisk(9 Q:)\Split IO/Sec
\PhysicalDisk(_Total)\Split IO/Sec

```

上面的信息也可以用作Typeperf命令的输入。添加一个-O参数并将输出信息写入到文本文件，如下所示：

```
typeperf -qx PhysicalDisk -o perf.txt
```

编辑该文本文件，使其只包含所要追踪的计数器。通过使用-Cf参数，其后跟随该计数器文件的文件路径，就可以使用该文件来确定追踪了哪些性能计数器。参考如下实例：

```
typeperf -cf perf.txt -o c:\perflogs\perf.bin -f bin
```

上面的命令中，Typeperf从Perf.txt中读取了要追踪的计数器列表，之后以二进制格式将性能数据写入到C:\perflogs目录下的文件中。

默认情况下，Typeperf以1秒为时间间隔对性能数据进行取样，除非按下Ctrl+C键。在使用命令行工作并监控输出时，这种做法是有益的。然而，如果同时还进行其他处理工作，不能对输出进行活跃监控时（大多数情况下是这样的），这种做法就不能起到好的效果。因此，通常需要控制取样间隔与持续时间。

要控制取样间隔与取样时间，可以分别使用-Si参数与-Sc参数。比如，如果希望Typeperf每隔120秒进行一次取样，并在取样100次之后终止，就可以使用如下命令：

```
typeperf -cf perf.txt -o c:\perflogs\perf.bin -f bin -si 120 -sc 100
```


管理员的一项重要工作就是监控网络系统，并确保一切运行正常或处于允许的范围内。第6章曾经讲过，密切关注事件日志有助于检测与追踪应用程序、系统安全以及基础服务方面存在的问题。通常，检测到或怀疑某处存在问题时，应该对其进行深入挖掘，以便找到导致问题的根源并加以纠正。但愿查明问题的根源，可以防止该问题再次发生。

7.1 管理应用程序、进程与性能

操作系统或用户启动服务、运行应用程序或执行命令时，Windows会启动一个或多个进程来处理相关的程序。有几款命令行工具可用于监控与管理程序，包括下面3个。

- **Task List (Tasklist)**。列出所有运行中进程的名称与进程ID，包括用户会话与内存使用等信息。
- **Task Kill (Taskkill)**。根据名称或进程ID终止运行中的进程。通过使用过滤器，也可以根据进程状态、会话号、CPU时间、内存使用情况、用户名以及其他信息来终止进程。
- **Powershell get-process**。显示性能数据的统计资料，包括CPU与内存使用情况、所有运行中进程列表等。用于获取资源使用情况与所有运行中进程的详细资料快照，只有在安装了Windows PowerShell之后才是可用的。

在接下来的内容中，将会详细讲解如何使用这些命令行工具。不过在具体讲解之前，先介绍一下进程的通常运行方式以及操作中常见的问题。

7.1.1 理解系统与用户进程

通常，由操作系统启动的进程称为系统进程，由用户启动的进程称为用户进程。大多数用户进程是以交互模式运行的，也就是说，用户启动进程时要与键盘或鼠标进行交互。如果应用软件或程序处于活跃状态并被选定，则相应的交互式进程就控制了键盘与鼠标，直至切换控制权（通过终止相应程序或选定其他程序）。当某进程具有控制权时，就称其为在“前台”运行。

进程可以独立于用户登录会话在后台运行，后台运行的进程不具备对键盘、鼠标或其他输入设备的控制权，通常由操作系统运行。通过使用任务计划程序，用户也可以将进程设定为在后台运行，并且这些进程的运行不受用户是否登录的影响。比如，如果任务计划程序在某用户登录时启动一个计划任务，该用户退出登录时，该进程会继续运行。

Windows对系统中运行进程的追踪是通过镜像名、进程ID、优先级以及其他一些记录资源使用情况的参数来实现的。镜像名是启动该进程的可执行程序的名称，比如Msdtc.exe或Svchost.exe。进程ID

是该进程的数字标识符，比如2588。进程ID还是一个优先级指示器，代表了该进程与其他运行中进程相比而言获取系统资源的优先程度。在进行优先级处理时，具有高优先级的进程可以比低优先级的进程更快地获取处理器时间、访问内存或操作文件系统。而低优先级的进程必须等待高优先级进程完成当前的处理任务之后，才可以访问CPU、内存或文件系统。

理想情况下，进程应该正确地按计划运行，而不出现任何问题。然而，实际的情况则是在你最不希望出问题的时候通常会出现问题。就进程而言，常见的问题包括下面3个。

- ❑ **进程失去响应。**比如，应用程序停止处理外部的请求。发生这种情况时，用户会报告无法访问特定的应用程序，提交的请求无法得到处理，或者被踢出该应用程序。
- ❑ **进程不释放CPU资源。**比如，某个失控进程几乎耗尽CPU资源。发生这种情况时，系统会变慢或失去响应，这是因为失控进程霸占了处理器时间，导致其他进程无法完成自己的任务。
- ❑ **进程占用的内存空间超过了正常的需求。**比如，应用程序发生了内存泄露。发生这种情况时，进程无法正常释放其所使用的内存资源，由此导致的结果是系统中可用内存逐渐下降，使得系统对外部响应变慢，或者失去响应。内存泄露还有可能导致系统中其他程序无法正常运行。

大多数情况下，发现系统进程存在上述问题或其他问题后，你可能需要终止该进程之后重启，并根据需要检查事件日志以便发现导致问题的根源。对内存泄露的情况，你可能需要向开发人员报告，以便发现是否有可用的更新程序。

此外，对存在内存泄露问题的应用程序，定期进行重启也是有用的。重启之后，操作系统会进行泄露内存的恢复。

7.1.2 检查运行中进程

如果需要检查本地或远程系统上运行的进程，可以使用Tasklist这一命令行工具。通过该工具，可以完成如下一些任务。

- ❑ 获取系统中运行进程的进程ID、状态以及其他一些重要信息。
- ❑ 查看运行中进程与系统中已配置服务的关系。
- ❑ 查看运行中进程使用的DLL。
- ❑ 使用过滤器包含或排除Tasklist查询获得的进程列表。

下面几节逐一讨论了这些任务。

1. 获取进程的详细信息

在本地系统上，通过在命令提示符中键入tasklist命令，就可以查看运行中任务列表。与很多其他的命令行工具类似，默认情况下，Tasklist以当前登录用户的许可权限运行。你也可以指定远程主机（要查询其上的任务）以及Run As许可权限，这是通过使用包含如下一些参数的扩展的语法格式实现的：

```
/s Computer /u [Domain\]User [/p Password]
```

其中，*Computer*为远程计算机名或IP地址，*Domain*为可选的域名，用户账号就存在于该域内，*User*为用户账号名（要使用的就是该用户账号的许可权限），*Password*为该用户账号的口令（可选）。如果没有指定域，则系统会假定当前域作为默认的域名。如果没有指定口令，则会弹出提示信息要求输入口令。

要了解如何添加计算机与用户信息，参考如下一些实例。

查询Mailer1上运行中任务

```
tasklist /s mailer1
```

查询192.168.1.5上使用账号adatum\wrstaneK运行的任务

```
tasklist /s 192.168.1.5 /u adatum\wrstaneK
```

上述命令的基本输出信息是以表格形式呈现的,通过分别使用/Fo List或/Fo Csv参数,也可以使得输出信息分别呈现为列表形式与逗号分隔的多行形式。你也可以使用重定向操作符(>或>>)将输出信息重定向到文件中,比如,tasklist /s mailer1>>current-tasks.log。

无论是本地系统还是远程系统,输出信息都应该类似于如下的格式:

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	28 K
System	4	Services	0	28,952 K
smss.exe	488	Services	0	776 K
csrss.exe	560	Services	0	5,272 K
wininit.exe	608	Services	0	4,056 K
csrss.exe	620	Console	1	13,004 K
services.exe	652	Services	0	7,456 K
lsass.exe	664	Services	0	1,852 K
Ism.exe	680	Services	0	6,400 K
svchost.exe	836	Services	0	7,228 K
winlogon.exe	868	Console	1	5,544 K
svchost.exe	932	Services	0	9,440 K
svchost.exe	984	Services	0	23,304 K
svchost.exe	1048	Services	0	12,208 K
svchost.exe	1100	Services	0	71,696 K
svchost.exe	1132	Services	0	36,920 K
dwm.exe	2832	Console	1	65,456 K
explorer.exe	2892	Console	1	25,624 K

输出信息中包含如下一些字段。

- 镜像名。进程或运行该进程的可执行程序镜像名。
- PID。进程的ID号。
- 会话名。运行该进程的会话名,如果取值为console则表明该进程是本地启动的。
- 会话#。运行该进程的会话标识符。
- 内存使用。在Tasklist运行时刻该进程使用的内存总量。

如果需要获取更多的详细信息,可以指定详细模式。这是通过包含/V参数实现的,使用该参数后,输出信息中会增加如下一些项目。

- Status。进程的当前状态,可以为运行、无响应或未知。状态为未知的进程仍然可以正常运行与响应,但处于无响应状态的进程通常必须终止或重启。
- 用户名。运行该进程的用户账号,并以域\用户格式列出。对由Windows启动的进程,用户账号通常为系统账号,比如系统、本地服务或网络服务等,域名则为NT AUTHORITY。
- CPU时间。该进程自启动到目前为止所使用的CPU时钟周期总量。
- 窗口标题。如果可用,则Windows显示进程名,否则为暂缺。比如,进程HelpPane.exe的进程

名显示为“Windows帮助和支持中心”。

使用Tasklist对运行中进程进行检查时，可能会发现两个独特的System与System Idle进程。System进程展示了本地系统进程的资源使用情况，System Idle进程则用于表述当前未使用的CPU资源。因而，如果System Idle进程的CPU一列取值为99，则说明当前有99%的系统资源处于闲置状态。如果觉得系统处于过载状态，则应该监控System Idle进程，查看CPU使用情况与总CPU时间。如果系统持续性地处于低空闲时间（即高CPU占用率），则可以考虑对处理器进行升级，甚而添加处理器。

检查进程时，要记住的一点是：单一的应用程序可能会启动多个进程。通常，这些进程依赖于一个主进程，并以该主进程为根形成一颗进程树。终止进程时，通常需要定位并终止该应用程序的主进程，而不是那些依赖性的附加进程，这将确保应用程序的彻底终止。

2. 查看运行中进程与服务的关系

运行Tasklist命令时，通过使用/Svc参数，可以检查运行中进程与系统中配置的服务之间的关系。输出信息中，包括进程镜像名、进程ID，以及使用该进程的所有服务的列表，类似于如下的格式：

Image Name	PID Services
System Idle Process	0 N/A
System	4 N/A
smss.exe	488 N/A
csrss.exe	560 N/A
wininit.exe	608 N/A
csrss.exe	620 N/A
services.exe	652 N/A
lsass.exe	664 KeyIso, ProtectedStorage, SamSs
lsass.exe	680 N/A
svchost.exe	836 DcomLaunch, PlugPlay
winlogon.exe	868 N/A
svchost.exe	932 RpcSs
svchost.exe	984 WinDefend
svchost.exe	1048 Audiosrv, Dhcp, Eventlog, Imhosts, wscntfrg
svchost.exe	1100 AudioEndpointBuilder, CscService, EMDMgmt, Netman, PcaSvc, SysMain, TabletInputService, TrkWks, UmRdpService, UxSms, WdiSystemHost, Wlansvc, WPDBusEnum, wudfsvc
svchost.exe	1132 AeLookupSvc, BITS, Browser, CertPropSvc, EapHost, gpsvc, IKEEXT, iphlpsvc, LanmanServer, MMCSS, ProfSvc, RasMan, Schedule, seclogon, SENS, SessionEnv, ShellHWDetection, Themes, Winmgmt, wuauserv
svchost.exe	1384 EventSystem, fdPHost, FDResPub, LanmanWorkstation, netprofm, nsi, SSDPSRV, upnphost, W32Time, WebClient
svchost.exe	1520 CryptSvc, Dnscache, KtmRm, NlaSvc, TapiSrv, TermService
spoolsv.exe	1776 Spooler
svchost.exe	1800 BFE, DPS, MpsSvc
dwm.exe	2832 N/A
explorer.exe	2892 N/A

默认情况下,上述命令的基本输出信息是以表格形式呈现的,并且不能使用`list`或`CSV`格式。除了格式之外,要注意的重要一点是服务是以缩略名形式列出的,这是`Sc`(一款用于对服务进行管理的命令行工具)的命名风格。

你可以使用进程与服务之间的关联关系来对系统进行管理。比如,如果感觉`W3svc`服务存在问题,对该服务进行故障排除时,一个有用的步骤就是监控该服务对应的进程。监控的主要要素包括:

- 进程状态;
- 内存使用;
- CPU时间。

通过对这些信息进行统计追踪,就有可能发现一些线索表明该进程已经停止响应,或者是一个霸占CPU资源的失控进程,或者存在内存泄露问题。

3. 查看进程使用的DLL

运行`Tasklist`命令时,通过使用`/M`参数,可以检查运行中进程与系统中配置的DLL之间的关系。输出信息中,包括进程镜像名、进程ID以及使用该进程的所有DLL的列表,如下面实例所示:

Image Name	PID	Modules
System Idle Process	0	N/A
System	4	N/A
smss.exe	488	N/A
csrss.exe	560	N/A
wininit.exe	608	N/A
csrss.exe	620	N/A
services.exe	652	N/A
lsass.exe	664	N/A
lsmd.exe	680	N/A
svchost.exe	836	N/A
winlogon.exe	868	N/A
svchost.exe	932	N/A
svchost.exe	984	N/A
svchost.exe	1048	N/A
svchost.exe	1100	N/A
svchost.exe	1132	N/A
dwm.exe	2832	ntdll.dll, kernel32.dll, ADVAPI32.dll, RPCRT4.dll, GDI32.dll, USER32.dll, msvcrt.dll, ole32.dll, OLEAUT32.dll, UxTheme.dll, IMM32.dll, MSCTF.dll, dwmredir.dll, SLWGA.dll, urlmon.dll, SHLWAPI.dll, iertutil.dll, WTSAPI32.dll, slc.dll, LPK.DLL, USP10.dll, comctl32.dll, NTMARTA.DLL, WLDAP32.dll, WS2_32.dll, NSI.dll, PSAPI.DLL, SAMLIB.dll, milcore.dll, dwmapi.dll, uDWM.dll, WindowsCodecs.dll, ctagent.dll, d3d9.dll, VERSION.dll, d3d8thk.dll, nvd3dum.dll, IconCodecService.dll
explorer.exe	2892	ntdll.dll, kernel32.dll, ADVAPI32.dll, RPCRT4.dll, GDI32.dll, USER32.dll, msvcrt.dll, SHLWAPI.dll, SHELL32.dll, ole32.dll, OLEAUT32.dll, SHDOCVW.dll,


```

UxTheme.dll, POWRPROF.dll, dwmapi.dll,
gdiplus.dll, slc.dll, PROPSYS.dll,
BROWSEUI.dll, IMM32.dll, MSCTF.dll,
DUser.dll, LPK.DLL, USP10.dll,
comctl32.dll, WindowsCodecs.dll,
apphelp.dll, CLBCatQ.DLL, cscui.dll,
CSCDLL.dll, CSCAPI.dll,
IconCodecService.dll, Secur32.dll,
rsaenh.dll, msiltsfcg.dll, VERSION.dll,
msi.dll, NTMARTA.DLL, WLDAP32.dll,
WS2_32.dll, NSI.dll, PSAPI.DLL, SAMLIB.dll,
SFC.DLL, sfc_os.dll, SETUPAPI.dll,
timedate.cpl, ATL.DLL, NETAPI32.dll,
OLEACC.dll, actxprxy.dll, USERENV.dll

```

获知进程加载了哪些DLL模块有助于确定导致进程不正常（失去响应、不释放CPU，或者占用了超过正常需求的内存）的原因。有些情况下，你可能需要检查DLL版本，以便确定其是否为系统应该使用的DLL版本。为此，你应该查询微软知识库或制造商文档，以便确定DLL版本及其他相关信息。

如果需要确定有哪些进程使用了某个指定的DLL，也可以在命令中指定DLL名。比如，如果怀疑打印机缓冲池驱动程序Winspool.drv是导致进程挂起的根源，就可以搜索使用Winspool.drv（而非Winspool32.drv）的进程，并检查这些进程的状态及资源使用情况。

用于指定DLL的语法格式如下：

```
tasklist /m DllName
```

其中，*DllName*为要搜索的DLL名。Tasklist在匹配DLL文件名时不区分大小写，你可以以任意大小写或混合形式输入要搜索的DLL文件名：

```
tasklist /m winspool.drv
```

通过上面的命令来搜索使用Winspool.drv的进程，其输出信息应该包含所有使用该DLL文件的进程名及各自的进程ID：

Image Name	PID	Modules
explorer.exe	2892	WINSPPOOL.DRV
rundll32.exe	3308	WINSPPOOL.DRV
acrotray.exe	3340	WINSPPOOL.DRV
IAAnotif.exe	3464	WINSPPOOL.DRV
IntelHCTAgent.exe	3584	WINSPPOOL.DRV
DrgToDsc.exe	3636	WINSPPOOL.DRV
WINWORD.EXE	4836	WINSPPOOL.DRV

4. 对Tasklist的输出进行过滤

通过Tasklist工具的/Fi参数，可以使用任何可用的信息字段对Tasklist的输出进行过滤，即便由于指定参数导致信息字段并没有正常包含在输出信息中。通过这种过滤机制，你可以只查看那些状态为失去响应的进程，或者只查看Svchost.exe进程相关的信息，或者查看占用了大量CPU时间的进程。

通过使用过滤器操作符，可以过滤特定的Tasklist信息字段，如下的一些过滤器操作符是可用的。

- **Eq**。等于，如果进程的某个字段包含了指定的值，则该进程应该包含在输出信息中。
- **Ne**。不等于，如果进程的某个字段包含了指定的值，则该进程应该排除在输出信息之外。

- Gt。大于，如果进程的某个字段包含了大于指定值的数值，则该进程应该包含在输出信息中。
- Lt。小于，如果进程的某个字段包含了小于指定值的数值，则该进程应该包含在输出信息中。
- Ge。大于等于，如果进程的某个字段包含了大于或等于指定值的数值，则该进程应该包含在输出信息中。
- Le。小于等于，如果进程的某个字段包含了小于或等于指定值的数值，则该进程应该包含在输出信息中。

如表7-1所示，过滤器操作符可以使用的值依赖于该操作符针对的Tasklist信息字段。要记住的是，所有字段都是可用的，即便某些字段不能根据指定的参数正常显示。比如，你可以对status字段进行匹配，而不使用/V（verbose）标记。

表7-1 过滤器操作符及其对应的有效Tasklist值

过滤器字段名	有效操作符	有效值
CPUTime	eq、ne、gt、lt、ge、le	以hh:mm:ss格式呈现的任意有效值
Services	eq、ne	任意有效的字符串
ImageName	eq、ne	任意有效的字符串
MemUsage	eq、ne、gt、lt、ge、le	以KB为计数单位的任意有效值
Modules	eq、ne	DLL名
PID	eq、ne、gt、lt、ge、le	任意有效的正整数
Session	eq、ne、gt、lt、ge、le	任意有效的会话编号
SessionName	eq、ne	任意有效的字符串
Status	eq、ne	运行、无响应、未知
Username	eq、ne	任意有效的用户名，只包括用户名，或者以域\用户的格式
WindowTitle	eq、ne	任意有效的字符串

要注意的是，必须使用双引号对过滤器字符串进行封装。参考如下实例，了解如何使用过滤器。寻找失去响应的进程：

```
tasklist /fi "status eq not responding"
```

对远程系统进行操作时，不能根据进程状态或窗口标题对进程进行过滤。有时候需要这样做，比如通过FIND命令进行管道输出：**tasklist /v /s Mailer1 /u adatum\wrstanek | find /i "not responding"**。要注意在这一场景中，待过滤字段必须存在于输出信息中，这也是使用/V参数的原因。进一步地，还可以使用/I参数使得FIND命令忽略字母大小写。

在Mailer1上搜索占用CPU时间超过30分钟的进程：

```
tasklist /s Mailer1 /fi "cputime gt 00:30:00"
```

在Mailer1上搜索占用内存超过20,000KB的进程：

```
tasklist /s Mailer1 /u adatum\wrstanek /fi "memusage gt 20000"
```

使用多个/Fi参数，使得输出信息必须同时匹配多个过滤器：

```
tasklist /s Mailer1 /fi "cputime gt 00:30:00" /fi "memusage gt 20000"
```

7.1.3 监控系统资源使用情况与进程

操作进程时，你通常可能需要获取系统资源使用情况的快照，这会准确展示内存的使用情况。获取快照的方法是使用Typeperf命令，来显示内存对象的关键计数器的当前值。如第6章中所讨论的，内存对象是大量性能对象中的一种，通过在命令行中键入typeperf -q Memory命令，就可以列出其相关的性能计数器。

表7-2总结了内存对象中的关键计数器，内存对象的大多数计数器显示的是上一次观测值或当前的百分数，而不是平均值。

表7-2 内存对象的关键计数器

内存对象计数器	计数器描述
%Committed Bytes In Use	提交字节占Commit Limit的比率。已提交的虚拟内存实际上是使用中的物理内存，这部分物理内存已经在页面文件中保留了相应空间，以便在需要写入磁盘时使用。Commit Limit是由页面文件的大小决定的，如果Windows增加了页面文件大小，则Commit Limit也会增加，而这一比率值会随之降低
Available MBytes	当前可用的物理内存总量（以MB计数），在数值上等于分配给备用（缓存的）、空闲、零页列表的内存总和，可以立即分配给进程或系统使用。当可用内存小于5%时，系统会变慢，性能也会受到不利影响
Cache Bytes	缓存字节数，在数值上等于系统缓存主驻留字节、系统驱动程序驻留字节、系统代码驻留字节以及换页池驻留字节等计数器的总和。它提供了操作系统内核使用的内存的相关信息。内核内存的关键部分必须存在于物理内存中，而不能在虚拟内存中，其他普通部分可以分布在虚拟内存中
Cache Bytes Peak	缓存字节数峰值，系统重启以来文件系统缓存使用的最大字节数
Cache Faults/sec	每秒缓存失效次数，在文件系统缓存中无法找到某内存页面，而必须从内存（软失效）或磁盘（硬失效）中其他位置寻找时，称之为缓存失效。文件系统缓存是一块物理内存区域，用于存储应用程序近期使用的页面数据
Commit Limit	以字节计数的虚拟内存总量，无需扩展页面文件就可以提交。随着提交字节数的增多，页面文件可以增加到最大值，此数值可以通过由commit limit减去物理内存总量而获取。如果初始页面文件设置过小，则系统会频繁地扩展页面文件，从而消耗了系统资源。因而，将初始页面大小设置为典型使用情况下的适当值（或者使用固定的页面文件大小）是必要的
Committed Bytes	已提交的虚拟内存总量（以字节计数）。已提交的虚拟内存实际上是使用中的物理内存，这部分物理内存已经在页面文件中保留了相应空间，用来在需要写入磁盘时使用。每个物理驱动器可以包含一个或多个页面文件，如果系统使用过多的虚拟内存（相对于系统中物理内存总量），则可能需要添加物理内存
Demand Zero Faults/sec	零化页面被请求（以满足失效时的需求）的速率，在数值上等于上两次采样的观测值除以采样间隔。对页面上存储数据进行清空并用0进行填充是Windows的一个安全特性，可以防止进程查看前面进程使用该页面存储的数据
Free&Zero Page List Bytes	分配给Free&Zero Page List的物理内存总量（以字节计数）。这块内存不包含缓存数据，可以立即分配给进程或系统使用
Free System Page Table Entries	系统当前未使用的空闲页面表入口数量
Modified Page List Bytes	分配给修改的页面列表的物理内存总量（以字节计数）。修改的页面列表的内存区域包含了进程、系统或系统缓存当前没有活跃使用的缓存数据与代码，在这块内存区域可以分配给进程或系统之前，Windows不能向其中写入数据

(续)

内存对象计数器	计数器描述
Page Faults/sec	每秒中页面失效次数, 包括硬失效与软失效。软失效后从内存中继续查询, 硬失效后则需要访问磁盘
Page Reads/sec	每秒钟为解决硬页面失效而需要的读操作次数。在请求的页面不存在于内存中而必须读取磁盘获取时, 就会发生硬页面失效。过多的硬失效会导致极大的延迟, 影响系统性能
Page Writes/sec	每秒钟页面文件写入到磁盘 (以便释放物理内存空间) 的次数, 只有在页面内容变化时, 才需要写入物理内存
Pages Input/sec	从磁盘读取页面 (以便解决硬页面失效) 的速度。在请求的页面不存在于内存中而必须读取磁盘获取时, 就会发生硬页面失效。过多的硬失效会导致极大的延迟, 影响系统性能
Pages Output/sec	页面写入磁盘 (以便释放物理内存空间) 的速度。如果计算机过于频繁地释放内存, 则说明系统物理内存 (RAM) 不足
Pages /sec	每秒钟写入磁盘与从磁盘读出的页面数。数值上是页面输入/每秒与页面输出/每秒的和
Pool Non-paged Allocs	对非换页池中空间进行分配的次数, 非换页池是一块用于不能写入磁盘的对象的系统内存区域, 一经分配就必须存在于物理内存之中
Pool Non-paged Bytes	非换页池的大小 (以字节计数), 非换页池是一块用于不能写入磁盘的对象的系统内存区域, 一经分配就必须存在于物理内存之中。如果相对于计算机中虚拟内存总量, 非换页池比较大, 则应该提高虚拟内存大小。如果非换页池大小随时间逐步减小, 则说明内核模式进程中可能存在内存泄露问题
Pool Paged Allocs	在换页池中为分配空间进行的调用的次数。换页池是一块系统内存区域, 用于存储那些在不使用时可写入磁盘的对象
Pool Paged Bytes	换页池的总容量 (以字节计数)。换页池是一块系统内存区域, 用于存储那些在不使用时可写入磁盘的对象。如果换页池容量相对于物理内存总量而言过大, 则可能需要为系统添加内存。如果换页池容量随时间逐步增大, 则可能内核模式进程存在内存泄露问题
Pool Paged Resident Bytes	当前驻留并处于活跃使用状态的换页池容量 (以字节计数)。典型情况下, 换页池中驻留字节数在数值上小于分配给换页池的总字节数
Standby Cache Core Bytes	分配给核心备用缓存页面列表的物理内存总量 (以字节计数), 备用缓存页面列表是一块内存区域。其中包含了当前没有被进程、系统或系统缓存活跃使用的缓存数据与代码, 可以立即分配给进程或系统使用。如果系统用完可用的 free-and-zero 内存区域, 则低优先级备用缓存页面列表上的内存区域将被重新规划在高优先级备用缓存页面列表上的内存区域之前
System Cache Resident Bytes	文件系统缓存中可分页的操作系统代码大小 (以字节计数)。该值只包含当前的物理页面, 不包含当前未驻留的任何虚拟内存页面
System Code Resident Bytes	当前存在于物理内存中的操作系统代码量 (以字节计数), 不使用时可写入磁盘
System Code Total Bytes	当前存在于虚拟内存中的可分页的操作系统代码大小 (以字节计数)。该值用于度量操作系统使用的物理内存 (不使用时可以写入磁盘) 总量, 不包括那些必须驻留在物理内存中 (而不能写入磁盘) 的代码
System Driver Resident Bytes	设备驱动程序当前使用的可分页物理内存大小 (以字节计数)。该部分内存区域是设备驱动程序的 working set (工作集, 物理内存区域)

(续)

内存对象计数器	计数器描述
System Driver Total Bytes	设备驱动程序当前使用的可分页物理内存与虚拟内存大小（以字节计数），包括写入到磁盘的物理内存、代码与数据
Transition Faults/sec	页面失效被解决的速度，解决的方法包括：通过由其他进程共享的，或存在于修改的页面列表或备用列表中的，或发生页面失效时正在被写入磁盘的页面进行恢复。页面恢复中不涉及到其他附加的磁盘活动
Transition Pages Repurposed/sec	Transition缓存页面被重用于其他目标的速度，如果不进行重用，这些页面本应保存于页面缓存中，重用之后，在被访问时导致软失效。这些页面可以包含私有内存或共享内存
Write Copies/sec	页面失效的速度，这里所说的页面失效特指某些写操作，这些写操作已经被系统通过从物理内存中其他位置进行页面复制而满足

示例7-1展示了如何使用Typeperf获取内存使用情况的快照。该实例中，使用了一个名为Perf.txt的文件来执行需要追踪的计数器，并以30秒为间隔进行了5次采样，采样结果存储到一个名为SaveData.txt的文件中。如果将此文件导入到电子表格中，或将其转换为Word文档中的表格，就会对输出内容有更好的感知，并对计算机使用内存的方式有准确的理解。

注解 选择追踪这些计数器是因为可以获取内存使用情况的一个整体快照，如果将该命令存储为脚本，就可以将其作为一个计划任务，并在每天的不同时间段获取当时内存使用情况快照。

示例7-1 获取内存使用情况快照

```
Command-line
typeperf -cf c:\logs\perf.txt -o c:\logs\savedata.txt -sc 5 -si 30
```

```
Source for Perf.txt
\memory\% Committed Bytes In Use
\memory\Available MBytes
\memory\Cache Bytes
\memory\Cache Bytes Peak
\memory\Committed Bytes
\memory\Commit Limit
\memory\Page Faults/sec
\memory\Pool Nonpaged Bytes
\memory\Pool Paged Bytes
```

```
Sample output
"(PDH-CSV 4.0)","\\SERVER12\memory\% Committed Bytes In Use ","
\\SERVER12\memory\Available MBytes", "\\SERVER12\memory\Cache Bytes", "
\\SERVER12\memory\Cache Bytes Peak", "\\SERVER12\memory\Committed Bytes", "
\\SERVER12\memory\Commit Limit", "\\SERVER12\memory\Page Faults/sec", "
\\SERVER12\memory\Pool Nonpaged Bytes", "\\SERVER12\memory\Pool Paged
Bytes"

"03/25/2008 14:24:28.033","22.860837","2023.000000","260632576.000000","
280514560.000000","1636175872.000000","7157112832.000000","80.494007","
73240576.000000","152875008.000000"
```



```
"03/25/2008 14:24:30.033","22.861294","2023.000000","260653056.000000","280514560.000000","1636208640.000000","7157112832.000000","70.997253","73240576.000000","152875008.000000"
"03/25/2008 14:24:32.033","22.861294","2023.000000","260653056.000000","280514560.000000","1636208640.000000","7157112832.000000","3.000142","73261056.000000","152875008.000000"
"03/25/2008 14:24:34.033","22.861581","2023.000000","260673536.000000","280514560.000000","1636229120.000000","7157112832.000000","15.999741","73154560.000000","152875008.000000"
"03/25/2008 14:24:36.033","22.861695","2023.000000","260681728.000000","280514560.000000","1636237312.000000","7157112832.000000","6.499981","73134080.000000","152875008.000000"
```

通过使用Windows PowerShell Get-Process cmdlet, 可以获取运行中进程的详细信息。表7-3总结了这一cmdlet的一些关键属性。在Windows PowerShell提示符下, 通过如下步骤, 可以查看所有进程的重要统计信息。

(1) 获取运行在服务器上所有进程, 并将其存储在\$a变量中, 这是通过如下命令实现的:

```
$a = get-process
```

(2) 使用InputObject参数, 将存储在\$a变量中的进程对象属性列表传递给format-table cmdlet, 这是通过如下命令实现的:

```
get-process -inputobject $a | format-table -property ProcessName,
BasePriority, HandleCount, Id, NonpagedSystemMemorySize,
PagedSystemMemorySize, PeakPagedMemorySize, PeakVirtualMemorySize,
PeakWorkingSet, SessionId, Threads, TotalProcessorTime,
VirtualMemorySize, WorkingSet, CPU, Path
```

在逗号分隔的属性列表中, 属性的顺序决定了其显示顺序。如果需要改变显示顺序, 只需要在该列表中将某属性移动到其他位置。

实际上, 如果确定了需要检查的进程, 并不需要使用这种多步骤的复杂过程。只需要输入进程名(不带.exe或.dll后缀), 而不需要使用-inputobject \$a。下面的实例中, 列出了winlogon进程相关的详细资料:

```
get-process winlogon | format-table -property ProcessName, BasePriority,
HandleCount, Id, NonpagedSystemMemorySize, PagedSystemMemorySize,
PeakPagedMemorySize, PeakVirtualMemorySize, PeakWorkingSet, SessionId,
Threads, TotalProcessorTime, VirtualMemorySize, WorkingSet, CPU, Path
```

你也可以输入进程名的一部分, 并使用*作为通配符来匹配进程名。下面的实例中, Get-Process列出了进程名以winl开始的所有进程:

```
get-process winl* | format-table -property ProcessName, BasePriority,
HandleCount, Id, NonpagedSystemMemorySize, PagedSystemMemorySize,
PeakPagedMemorySize, PeakVirtualMemorySize, PeakWorkingSet, SessionId,
Threads, TotalProcessorTime, VirtualMemorySize, WorkingSet, CPU, Path
```

提示 默认情况下, 很多用于度量内存使用情况的属性是以32位值的形式定义的。在64位系统上操作Get-Process时, 你会发现这些属性值同时具备32位与64位两个版本。在内存大于4GB的64位系统上, 需要使用64位的版本, 以便获取准确的属性值。

表7-3 Get-Process的属性及其使用方式

属性名	属性描述
BasePriority	展示进程的优先级。优先级决定了系统资源分配给该进程的优先程度，标准的优先级包括：低（4）、低于标准（6）、标准（8）、高于标准（10）、高（13）、实时（24）。默认情况下，大多数进程的优先级为标准，最高的优先级一般赋予实时进程
CPU	展示该进程的CPU占用率。System Idle进程展示了CPU资源有多大百分比是空闲的，如果值为99，则说明当前有99%的系统资源处于空闲状态。如果在峰值或平均使用率上系统空闲时间较低（意味着高CPU使用率），则可以考虑到升级到更快的处理器或添加处理器
Description	展示进程的描述信息
FileVersion	展示进程可执行文件的文件版本
HandleCount	展示由该进程维护的文件句柄数，该数值描述了该进程对文件系统的依赖程度。有些进程有数千个打开的文件句柄，每个文件句柄都需要一定的系统内存进行维护
Id	展示该进程的运行时标识数
MinWorkingSet	展示该进程使用的working set内存的最小总量
Modules	展示该进程使用的可执行文件与动态链接库
NonpagedSystemMemorySize/ onpagedSystemMemory Size64	展示该进程所拥有的、不允许写入磁盘的虚拟内存总量。这块RAM内存区域也就是非换页池，用于存储那些不能写入到磁盘的对象，对需要很高非换页池容量的对象，应该重点关注。如果服务器没有足够的空闲空间，则这些进程很可能会成为导致大量页失效的根源
PagedSystemMemorySize/ agedSystemMemorySize64	展示该进程所提交的、可以写入磁盘的虚拟内存总量。这块RAM内存区域也就是换页池，用于存储那些可以在不使用时写入到磁盘的对象。随着进程的不断活动，所需要的内存缓冲池容量也不断增加，大多数进程对换页池的需求超过对非换页池的需求
Path	展示该进程可执行文件的全路径
PeakPageMemorySize/ PeakPageMemorySize64	展示该进程使用的页面内存总量的峰值
PeakVirtualMemorySize/ PeakVirtualMemorySize64	展示该进程使用的虚拟内存总量的峰值
PeakWorkingSet/ PeakWorkingSet64	展示该进程使用的内存总量的最大值，包括私有的working set与非私有的working set。如果峰值内存异常大，则可能存在内存泄露问题
PriorityBoostEnabled	显示一个布尔型值，用于表示进程是否激活了PriorityBoost特性
PriorityClass	显示进程的优先级类别
PrivilegedProcessorTime	显示该进程内核模式使用时间的总量
ProcessName	显示进程名
ProcessorAffinity	显示该进程的处理器关联设置
Responding	显示一个布尔型值，用于表示测试时该进程是否响应
SessionId	显示运行该进程的用户识别号（会话），对应的是任务管理器中用户选项卡的ID
StartTime	显示该进程启动的时间与日期

(续)

属 性 名	属性描述
Threads	显示该进程正在使用的线程数。大多数服务器应用程序都是多线程的,用于处理并发请求,有些应用程序还可以动态控制并发执行的线程数,用来提高应用程序的性能。然而,过多的线程也会降低性能,因为操作系统不得不进行频繁的线程上下文切换
TotalProcessorTime	显示进程启动以来使用的CPU时间总量。如果某进程使用了大量CPU时间,则相关的应用程序可能存在配置问题,也可能表示某个失控进程或失去响应的进程不必要地占用了CPU
UserProcessorTime	显示进程在用户模式下使用时间的总量
VirtualMemorySize/ VirtualMemorySize64	显示为某进程分配与保留的虚拟内存总量。虚拟内存是位于磁盘上的内存,在访问速度上慢于缓冲池内存。如果将应用程序配置为使用更多的物理内存,则有助于提高性能。当然,这样做的前提是系统中存在可用的内存,否则会导致系统中运行的其他进程性能下降
WorkingSet/ WorkingSet64	显示进程当前正在使用的内存总量,包括私有的working set与非私有的working set。私有的working set是指不能与其他进程共享的内存区域,非私有的working set则与其他进程共享。如果某进程的内存使用随时间缓慢增长,并且不再回退到正常值,则可能存在内存泄露问题

7.1.4 终止进程

如果需要终止运行在本地或远程系统上的进程,可以使用命令行工具Taskkill。通过该工具,可以根据进程ID(使用/Pid参数)或进程镜像名(使用/Im参数)来终止进程。如果需要根据进程ID或进程镜像名来终止进程,则可以输入多个/Pid参数与/Im参数。不过,在使用镜像名时要特别小心,因为Taskkill会终止所有使用该镜像名的进程。因而,如果系统中运行了Helpctr.exe的3个实例,则对该镜像名使用Taskkill之后,所有这3个进程都将被终止。

与Tasklist类似,默认情况下,Taskkill以当前登录用户的许可权限运行。你也可以指定远程主机(要查询其上的任务)以及Run As许可权限,这是通过使用包含如下一些参数的扩展的语法格式实现的:

```
/s Computer /u [Domain\]User [/p Password]
```

其中,Computer为远程计算机名或IP地址,Domain为可选的域名,用户账号就存在于该域内,User为用户账号名(要使用的就是该用户账号的许可权限),Password为该用户账号的口令(可选)。如果没有指定域,则系统会假定当前域作为默认的域名。如果没有指定口令,则会弹出提示信息要求输入口令。

注解 有时候,强制进程终止运行是必要的。典型情况下,当进程在打开文件、读写数据或执行其他读写操作的过程中停止响应时,就需要强制终止。要强制终止进程,可以使用/F参数,且该参数只适用于本地系统上运行的进程。远程系统上终止的进程通常都是强制终止的。

提示 检查进程时,要记住的是,一个的应用程序可能会启动多个进程。通常,这些进程依赖于一个主进程,并以该主进程为根形成一颗进程树。偶尔,你可能需要终止整个进程树,包括父应用程序进程与所有依赖于该进程的进程。要做到这一点,可以使用/T参数。

要了解如何使用Taskkill，参考如下一些实例。

终止进程ID为208的进程：

```
taskkill /pid 208
```

终止镜像名为cmd.exe的所有进程：

```
taskkill /im cmd.exe
```

终止Mailer1上进程ID为208、1346、2048的进程：

```
taskkill /s Mailer1 /pid 208 /pid 1346 /pid 2048
```

强制终止PID为1346的进程：

```
taskkill /f /pid 1346
```

终止进程树，以PID为1248的进程开始，包括所有子进程：

```
taskkill /t /pid 1248
```

要确保只终止匹配特定标准的进程，可以使用表7-1中列出的所有过滤器（SessionName除外）。比如，你可以使用过滤器规定只有失去响应的cmd.exe实例才应该终止，而不是终止所有的cmd.exe实例（使用/I参数时，默认情况下是终止所有）。

与Tasklist类似，Taskkill提供了模块过滤器以及EQ与NE等操作符，用于指定应该包含或排除的DLL。回想一下，Tasklist /M的作用是检查运行中进程与系统中配置的DLL之间的关系。通过使用Taskkill模块过滤器与EQ操作符，可以终止所有使用某特定DLL的进程。通过使用Taskkill模块过滤器与NE操作符，可以确保所有使用某特定DLL的进程不会被终止。

提示 使用过滤器时，并不一定需要指定特定的镜像名或进程ID。也就是说，可以只根据进程是否匹配过滤器标准来终止进程。比如，你可以规定只终止所有失去响应的进程。

与Tasklist类似，你可以在Taskkill中使用多个过滤器，要记住必须使用双引号封装过滤字符串。要了解如何在Taskkill中使用过滤器，参考如下实例。

终止失去响应的cmd.exe实例：

```
taskkill /im cmd.exe /fi "status eq not responding"
```

终止所有PID大于4并且失去响应的进程：

```
taskkill /fi "pid gt 4" /fi "status eq not responding"
```

终止所有使用Winpool.drv这一DLL的进程：

```
taskkill /fi "modules eq winpool.drv"
```

可以看到，尽管第2个实例中没有使用/Im参数与/Pid参数，但对进程ID进行了过滤，从而只影响部分进程。你可能需要避免无意间终止System进程或System Idle进程。典型情况下，这两个进程的PID

分别为4与0。如果终止了这两个进程，系统就会停止响应或关机。

7.2 通过监控来检测与解决性能问题

在命令行中，Tasklist与Windows PowerShell Get-Process可以检测与解决大多数性能问题。不过，你还是需要进行深入研究，以便发现存在的问题。如果存在问题，就努力找出导致问题的根源。

7.2.1 监控内存分页与磁盘页面

通常，你可能需要获取关于内存页面硬失效与软失效的详细信息。当某进程请求内存中的一个页面，而这个页面不存在于请求的位置时，就会发生页面失效。如果请求的页面内容存在于内存中的某处，就称之为页面软失效。如果请求的页面内容只能从磁盘获取，就称之为页面硬失效。

如果需要实时查看系统中发生页面失效的情况，可以在命令行中输入如下命令：

```
typeperf "\memory\Page Faults/sec" -si 5
```

要停止Typeperf，可以按Ctrl+C。页面失效是根据每秒钟硬失效与软失效的次数展示的。其他可用于追踪页面失效的内存对象计数器包括：

- ❑ Cache Faults/sec;
- ❑ Demand Zero Faults/sec;
- ❑ Page Reads/sec;
- ❑ Page Writes/sec;
- ❑ Write Copies/sec;
- ❑ Transition Faults/sec;
- ❑ Transition Pages Repurposed/sec.

需要特别关注的是Page Reads/sec与Page Writes/sec这两个计数器，因为这两个计数器提供了页面硬失效的信息。尽管开发人员主要关心的是导致页面失效的根源，但管理员更关心的是发生了多少次页面失效。

大多数处理器都可以处理大量的软失效，即从内存中其他位置处寻找请求的内存页面。而对于硬失效，则必须从磁盘中取回请求的内存页面，这可能导致很大的延迟。如果系统中发生了大量的硬失效，则可能需要增加系统的内存总量，或者降低系统与应用程序缓存的内存总量。

除了前面讨论的计数器与内存对象之外，还可以使用下面的对象与计数器来检查磁盘页面问题。

- ❑ **Paging File(*)\%Usage**。当前使用中页面文件占总页面文件百分比。如果对所有实例，这一数值都接近100%，则应该考虑增大虚拟内存容量，或者为系统增添物理内存。这将确保计算机有充足的内存资源，以备不时之需（比如计算机工作负载增加的时候）。
- ❑ **Paging File(*)\%Usage Peak**。当前使用中页面文件占可用的总页面文件百分比的峰值。如果这一数值过高，则说明页面文件不够大，不足以处理增加的工作负载需求。
- ❑ **PhysicalDisk(*)\% Disk Time**。选定磁盘用于处理读写请求所花费的时间。对其上包含页面文件的物理磁盘，应该关注这一数值。如果这一数值在几个监测周期内呈增加趋势，则应该密切关注页面文件使用情况，并考虑为系统增添物理内存。
- ❑ **PhysicalDisk(*)\Avg. Disk Queue Length**。采样周期内等待选定磁盘处理的平均读写请求次数。对其上包含页面文件的物理磁盘，应该关注这一数值。如果这一数值呈增加趋势，同时

Memory\Page Reads/Sec也在增加, 则说明系统正在处理大量的页面文件读请求。

上面的描述中, 圆括号中的星号是对象实例的占位符。如果某特定对象有多个实例, 比如计算机有多个物理磁盘或多个页面文件, 则可以使用一个对象实例追踪该对象的特定出现。你也可以追踪某对象的所有实例, 比如监控某系统上所有物理磁盘。通过_Total, 可以操作所有计数器实例, 或指定待监控的单独计数器实例。

示例7-2展示了如何使用Typeperf获取磁盘页面快照, 该实例中, 使用了一个名为Perf.txt的文件来执行需要追踪的计数器, 并以30秒为间隔进行了5次采样, 采样结果存储到一个名为SaveData.txt的文件中。如果将此文件导入到电子表格中, 或将其转换为Word文档中的表格, 就会对输出内容有更好的了解, 并对计算机使用页面文件以及文件分页的方式有准确的理解。

示例7-2 检查磁盘页面

```
Command line
typeperf -cf c:\logs\pageperf.txt -o c:\logs\savepagedata.txt -sc 5
-si 30
```

```
Source for PagePerf.txt
\memory\Pages/Sec
\Paging File(_Total)\% Usage
\Paging File(_Total)\% Usage Peak
\PhysicalDisk(_Total)\% Disk Time
\PhysicalDisk(_Total)\Avg.Disk Queue Length
```

7.2.2 监控单个进程的内存使用与 Working Memory Set

你可以使用Tasklist获取某个进程的基本的内存使用情况, 其语法格式为:

```
tasklist /fi 'pid eq ProcessID'
```

其中, ProcessID为待操作进程的进程ID, 输出信息将展示该进程当前使用了多少内存资源。比如, 如果待追踪进程PID为7292, 则上面命令的输出类似于如下的格式:

Image Name	PID	Session Name	Session#	Mem Usage
jvappm.exe	7292		1	7,424 K

从输出信息可以看出, 该进程占用了7,424KB内存资源。通过对内存使用情况的监控, 可以判断占用的内存资源是否不断增长。如果某进程占用的内存资源超过了典型情况下应该占据的内存资源标准, 则该进程可能存在内存相关的问题。

示例7-3提供了一个命令行脚本源代码, 该脚本用于定期检查进程的资源使用情况。运行时, 该脚本要求以进程ID作为第一个参数。如果没有提供PID, 则会返回错误信息。

示例7-3 在命令行中查看内存使用情况

```
MemUsage.bat
@echo off
if "%1"=="" (echo Error: please enter Process ID to track) & (goto EXIT)
EXIT
```

```
tasklist /fi "pid eq %1"
timeout /t 600
tasklist /fi "pid eq %1"
timeout /t 600
tasklist /fi "pid eq %1"
: EXIT
```

Sample output

Image Name	PID	Session Name	Session#	Mem Usage
jvapi.exe	7292		1	7,452 K

Waiting for 0 seconds, press a key to continue ...

Image Name	PID	Session Name	Session#	Mem Usage
Jvapi.exe	7292		1	7,452 K

Waiting for 0 seconds, press a key to continue ...

Image Name	PID	Session Name	Session#	Mem Usage
jvapi.exe	7292		1	7,452 K

示例7-3中, 进程的内存使用没有随着不同的采样间隔而变化。因此, 该进程不太可能存在内存泄露问题, 但如果要确认这一点, 则需要更多次采样。

你可以使用Windows PowerShell `Get-Process` cmdlet来追踪某进程内存使用情况的详细信息, 其语法格式如下:

```
get-process ProcessName | format-table -property
NonpagedSystemMemorySize, PagedSystemMemorySize, VirtualMemorySize,
PeakVirtualMemorySize, MinWorkingSet, WorkingSet, PeakWorkingSet
```

其中, *ProcessName*为进程名(不带.exe或.dll后缀)。在Windows PowerShell脚本中, 如示例7-4, 可以将`Get-Process` cmdlet与`start-sleep` cmdlet结合起来, 以便在定时的时间间隔查看某进程的内存使用情况。

示例7-4 查看内存使用情况的详细信息

```
MemUsage.ps1
get-process msdtc | format-table -property NonpagedSystemMemorySize,
PagedSystemMemorySize, VirtualMemorySize, PeakVirtualMemorySize,
MinWorkingSet, WorkingSet, PeakWorkingSet

start-sleep -seconds 600

get-process msdtc | format-table -property NonpagedSystemMemorySize,
PagedSystemMemorySize, VirtualMemorySize, PeakVirtualMemorySize,
MinWorkingSet, WorkingSet, PeakWorkingSet

start-sleep -seconds 600

get-process msdtc | format-table -property NonpagedSystemMemorySize,
PagedSystemMemorySize, VirtualMemorySize, PeakVirtualMemorySize,
```

MinWorkingSet, WorkingSet, PeakWorkingSet						
Sample output						
Nonpaged System MemorySize	PagedSystem Memory Size	Virtual Memory Size	Peak Virtual MemorySize	Working Set	Peak Working Set	
6304	70544	41766912	63631360	6287360	6344704	
Nonpaged System MemorySize	PagedSystem Memory Size	Virtual Memory Size	Peak Virtual MemorySize	Working Set	Peak Working Set	
8123	96343	56243535	97423424	9147256	9348942	
Nonpaged System MemorySize	PagedSystem Memory Size	Virtual Memory Size	Peak Virtual MemorySize	Working Set	Peak Working Set	
17564	129645	48934246	97423424	9987384	10344706	

注解 Windows PowerShell脚本文件的扩展名为.ps1, 要在Windows PowerShell提示符中运行脚本, 需要键入脚本名以及可选的扩展名。要注意的是, 必须指定脚本文件的全路径, 即便脚本文件在当前目录下。要指定当前目录, 可以使用目录名或圆点符号(.). 比如, 对于当前目录下的MemUsage.ps1文件, 可以在Windows PowerShell提示符中键入.\memusage.ps1来运行该脚本。

示例7-4中, Get-Process提供了如下一些信息。

- ❑ NonPagedSystemMemorySize。展示了不能写入磁盘的已分配内存总量。
- ❑ PagedSystemMemorySize。展示了可以与磁盘进行页面交换的已分配内存总量。
- ❑ VirtualMemorySize。展示了为某进程分配与保留的虚拟内存总量。
- ❑ PeakVirtualMemorySize。展示了某进程使用的虚拟内存总量峰值。
- ❑ WorkingSetSize。展示了操作系统分配给进程的内存总量。
- ❑ PeakWorkingSet。展示了进程使用的内存总量峰值。

关注这些属性时, 实际上是聚焦于某个特定进程的内存使用情况。要监控的关键内容是working memory set, 指的是操作系统为该进程分配的内存。如果该数值随时间增长, 而最终不能回退到正常情况下的标准值, 则该进程可能存在内存泄露问题。发生内存泄露时, 进程无法正常释放占用的内存, 从而导致系统整体性能下降。

示例7-4中, 进程的内存使用随着采样间隔不断增加, 最可能的原因是该进程正被用户或计算机本身活跃地使用。正常的话, 该进程的内存使用会回归到一个标准值。如果不能, 则说明该进程可能存在内存相关的问题。

7.2.3 解决性能瓶颈

由于内存是工作站与服务器上的主要性能瓶颈, 本章前面讨论的很多技术都有助于确定系统中存在的内存相关问题。在系统不能正常运行时, 内存应该是第一个检查的目标。

然而，内存并不是唯一的性能瓶颈，处理器也可能会成为系统性能的瓶颈。比如，进程的多线程需要的CPU时间超过了可用的CPU资源。如果处理器成为系统性能瓶颈，则增加内存、驱动器或网络带宽都不能解决问题。只能对处理器进行升级（更快的主频），或者添加处理器，以便提高计算机的性能上限。对于服务器，也可以将处理器密集型的应用程序转移到其他服务器上运行。

可用于检查处理器瓶颈的Typeperf计数器包括如下5个。

- **System\Processor Queue Length**。记录等待执行的线程数。这些进程在一块由系统中所有处理器共享的区域中排队等待执行，由于线程在执行之前必须等待。因此，处理器队列是逐渐增长的，从而导致系统响应变慢，甚至失去响应。为避免这种情况，需要对处理器进行升级（更快的主频），或者添加处理器，来提高服务器的性能上限。
- **Processor(*)\% Processor Time**。记录选定的处理器执行非空闲进程的时间百分比。对服务器上的每个处理器实例，应该单独追踪这一计数器。如果对所有处理器实例，% Processor Time值都很高（高于75%），而同时网络吞吐率与磁盘吞吐率又较低，则需要对处理器进行升级（更快的主频），来提高服务器的性能上限。
- **Processor(*)\% User Time**。记录选定的处理器执行用户模式下非空闲进程的时间百分比。用户模式是一种用于应用程序与用户级子系统的处理模式。如果对所有进程实例，这一取值都很高，则表明需要对处理器进行升级（更快的主频），或者添加处理器，来提高服务器的性能上限。
- **Processor(*)\% Privileged Time**。记录了选定的处理器用于特权模式下执行非空闲线程时间的百分比。特权模式是一种用于操作系统组件与服务的处理模式，可以直接对硬件与内存进行访问。如果对所有进程实例，这一取值都很高，则表明需要对处理器进行升级（更快的主频），或者添加处理器，来提高计算机的性能上限。
- **Processor(*)\Interrupt/sec**。记录了选定处理器接收与处理硬件中断的平均速度（事件数/秒）。如果这一数值随时间不断增加，但实际上的系统活动并没有真正的相应增加，则说明系统硬件可能存在问题。要解决这一问题，必须找到导致这一问题的设备或组件。驱动程序或磁盘子系统（比如硬盘驱动器或网络组件）产生中断后，处理器必须停止当前工作来处理中断请求，因为来自硬件的中断请求具有更高的优先级。然而，设计上存在缺陷的驱动程序或系统组件可能会产生假中断请求，这些假中断请求会浪费大量的CPU资源。系统主板或失效的组件也会产生假中断请求。

注解 上面的描述中，圆括号中的星号是对象实例的占位符。在多处理器系统上，你可能需要取消处理器关联，避免处理器成为系统性能瓶颈。通过处理器关联，可以将程序或进程设定为由特定的处理器运行，来提高其性能，但处理器关联也可能会导致其他程序或进程难于访问与使用该处理器。

系统硬盘极少成为系统性能瓶颈的主要原因。如果系统不得不进行大量的磁盘读、写操作，通常是因为缺少足够的可用物理内存，使得系统不得不通过磁盘进行页面读写操作。由于磁盘的读写远远慢于内存的读写，因此，过多的磁盘页面读写会降低系统的整体性能。要减少磁盘活动总量，就需要将系统的内存管理策略设置得尽可能高效，使其只在必要的时候才对磁盘进行页面读写。

你可以使用下面的计数器对磁盘读写进行监控。

- **PhysicalDisk(*)\Disk Time**。记录了物理磁盘处于忙碌状态的百分比。对系统中所有硬盘驱动器,可以联合追踪这一数值与Processor(*)\%Processor Time、Network Interface(*)\Bytes Total/sec等值。如果% Disk Time值较高,而处理器与网络连接等相关数值不高,则说明硬盘驱动器可能已经成为系统性能瓶颈。
- **PhysicalDisk(*)\Current Disk Queue Length**。记录了当前等待磁盘访问的请求数。如果此数值较高,则说明磁盘等待影响了系统性能。通常,等待的请求数越少越好。
- **PhysicalDisk(*)\Avg. Disk Write Queue Length**。记录等待处理的写请求数。
- **PhysicalDisk(*)\Avg. Disk Read Queue Length**。记录等待处理的读请求数。
- **PhysicalDisk(*)\Disk Writes/sec**。记录每秒钟磁盘写操作次数,该数值代表了磁盘I/O总量。通过追踪每秒钟的磁盘写操作次数以及写队列的长度,可以确定写操作对磁盘性能的影响。
- **PhysicalDisk(*)\Disk Reads/sec**。记录每秒钟磁盘读操作次数,该数值代表了磁盘I/O总量。通过追踪每秒钟的磁盘读操作次数以及读队列的长度,可以确定读操作对磁盘性能的影响。

网络组件也可能成为系统性能的瓶颈。用户发起请求、请求得到处理、用户获取响应信息之间的延迟都会让用户感觉系统响应很慢。遗憾的是,在很多场景下,用户在网络操作中感觉到的延迟并不是管理员所能控制的,这是因为造成延迟可能与用户网络连接的类型与请求信息的路由方式有关。当然,计算机处理网络请求的总容量以及可用带宽总量是可以控制的。网络容量是由配置在计算机上的网卡与网络接口决定的,可用的网络带宽则取决于网络基础设施与请求发生时其上的网络流量。

你可以使用如下的计数器来检查网络活动并寻找瓶颈。

- **Network Interface(*)\Bytes Received/Sec**。用于记录网络适配器接收字节的速度。
- **Network Interface(*)\Bytes Sent/Sec**。用于记录网络适配器发送字节的速度。
- **Network Interface(*)\Bytes Total/Sec**。用于记录网络适配器发送、接收字节的速度。如果感觉存在问题,可以检查网卡配置。
- **Network Interface(*)\Current Bandwidth**。用于评估选定网络适配器当前带宽(位/秒)。可以通过检查发现当前带宽是否与计算机上配置的网卡类型匹配。大多数计算机使用10M、100M或1G的网卡。要记住的是,如果计算机上安装了1G的网卡,则该计算机连接的网络设备必须支持这一速度,才能发挥最大效能。



对管理员来说，未雨绸缪地管理与监控系统是一项重要的工作内容。遗憾的是，大多数管理员没有时间对自己负责的所有系统进行有规律的例行维护与监控。有鉴于此，本章将深入挖掘系统维护、监控的自动化方法与技术，通过这些方法与技术，可以降低管理员的工作负担，节省管理时间。

在前面的相应章节中，已经讲解了如何追踪与使用事件日志、如何进行自动化监控、如何监控进程以及如何对性能问题进行故障排除等内容。本章将讲解关于事件记录方式、企业级集中化事件记录、收集与生成性能数据报告等技术。管理员与其他具备相应许可权限的用户可以阅读事件日志、配置事件日志、管理性能日志等，本章讲解的技术将有助于完成这些工作。

8.1 管理事件日志

通常，管理员需要对本地及远程系统上的事件日志配置进行管理，以便确保事件日志以期望的方式进行了配置。比如，组织的安全策略可能会要求确保安全日志不能被重写。根据不同的配置需求，你可以使用命令行或命令行脚本来对安全日志或其他事件日志进行精确配置，而不需要登录每一台需要配置的本地或远程系统。配置事件日志时，首选的工具是Windows事件命令行工具（Wevtutil），你可以使用Wevtutil来完成查看或修改事件日志的配置、读取事件、导出与存档事件日志、清除事件日志等任务。

8.1.1 开始使用 Wevtutil

Wevtutil提供了大量的命令与选项，可用于管理事件日志及其配置。在使用这一工具之前，应该花一点时间了解一下可用的子命令与选项。子命令与选项都有短格式的缩略名与长格式的完整名。如果刚开始接触Wevtutil，或者希望脚本中使用的命令更加清晰易读，你可能需要使用长格式的完整名，以便减少可能的混淆与误解。熟练之后，一般会使用短格式的缩略名，以便节省时间与键盘输入。

Wevtutil的基本语法格式为：

```
wevtutil Command Argument [[Argument]...] [/Option: Value [/Option:Value]...]
```

其中，*Command*是表8-1中列出的某条命令，*Arguments*是命令参数，*Value*是某选项的值。

表8-1 Wevtutil中可用的命令

长 格 式	短 格 式	描 述
archive-log	Al	对导出的日志进行存档
clear-log	Cl	清空事件日志，永久性删除其中的所有事件

(续)

长 格 式	短 格 式	描 述
enum-logs	El	根据名称列出所有可用的日志
enum-publishers	Ep	列出所有已注册的事件发布者，包括所有的Windows服务，以及其他配置为向事件日志中写入事件的组件
export-log	Epl	以Windows事件日志格式(.evtx)导出日志
get-log	Gl	获取日志的配置信息
get-log-info	Gli	获取日志的状态信息
get-publisher	Gp	获取事件发布者的配置信息
install-manifest	lm	从清单安装事件发布者与日志
query-events	Qe	从日志或日志文件查询事件
set-log	Sl	修改日志的配置
uninstall-manifest	Um	从清单卸载事件发布者与日志

尽管上面每条Wevtutil命令的选项集有些微不同，但所有命令有一些通用的选项，如表8-2中所示。默认情况下，Wevtutil以当前登录用户的权限运行，但也可以指定在远程计算机上运行Wevtutil命令的Run As权限。要做到这一点，需要使用扩展的语法格式，其中包含如下的参数：

```
/r:Computer /u:[Domain\]User [/p:Password]
```

其中，*Computer*是远程计算机名或IP地址，*Domain*是可选的域名（用户账号存在于该域中），*User*是用户账号名（要使用的就是该用户账号的许可权限），*Password*是该用户账号的口令（可选的）。如果不指定域，则系统会将当前域设定为要使用的域。如果没有提供口令，则系统会弹出提示要求输入口令。

注解 选项与选项值是由冒号分隔开的，且冒号后不要插入空格。

表8-2 Wevtutil命令选项

长 格 式	短 格 式	描 述
/remote:	/r:	指定要在其上运行命令的远程计算机，注意im（install-manifest）与um（uninstall-manifest）中不能使用该选项
/username:	/u:	指定一个以域\用户或用户方式登录远程计算机的不同用户，只有在使用远程计算机时才是可用的
/password:	/p:	为指定的用户设置口令。如果未提供口令，或者使用*作为口令，系统会弹出提示要求输入口令。只有在使用远程计算机，并且使用一个不同的用户登录时，这一选项才是可用的
/authentication:	/a:	设置连接到远程计算机时的认证类型，可以设置为Default、Negotiate、Kerberos或NTLM。默认设置为Negotiate
/unicode:	/uni:	将输出显示设置为Unicode或ASCII文本。设置为真时，代表Unicode；设置为假时，代表ASCII文本。默认设置为ASCII文本

8.1.2 列出可用的日志与已注册的事件发布者

前面第6章中曾经说过，系统中有哪些可用的事件日志取决于系统中已安装的角色与服务。对已

注册的事件日志发布者来说同样如此，一般包括所有Windows服务以及配置为向事件日志中写入事件的组件。

通过在命令行中使用`wevtutil el`命令，可以列出计算机中所有可用的日志。由于这种方式得到的日志列表太长，以至于无法在命令提示符中方便地显示。因此，你可能需要使用`FIND`命令对输出信息进行重定向，以便发现特定的日志，下面给出的就是这样一个实例：

```
wevtutil el | find /i "FindText"
```

其中，*FindText*是要搜索的文本，用于对输出信息进行查找和过滤，以便发现特定的日志。比如，如果希望确定某特定日志是否与加密文件系统（EFS）关联，就可以使用如下命令对输出信息进行过滤：

```
wevtutil el | find /i "efs"
```

由于使用了`/i`参数，`FIND`命令会忽略输出文本中字母的大小写。因而，包含EFS（大写、小写的任意组合）关键字的日志名都会被选中，比如：

```
Microsoft-Windows-EFS/Debug
```

通过在命令行中使用`wevtutil ep`命令，可以列出计算机中所有已注册的事件日志发布者。同样地，由于这种方式得到的输出列表太长，你可能需要使用`FIND`命令对输出信息进行过滤。比如，如果想确定Netlogon服务是否是一个已注册的事件日志发布者，就可以在命令提示符中输入如下命令：

```
wevtutil ep | find /i "netlogon"
```

由于Windows服务、系统组件以及其他附加软件都可能注册为事件日志发布者，你可能希望了解事件发布者向事件日志中写入事件的确切方式。为此，你可以使用`Wevtutil gp`命令并以如下的语法格式列出事件发布者的配置信息：

```
wevtutil gp Publisher
```

其中，*Publisher*为所要使用的、已注册的事件日志发布者的全名。比如，在确定Netlogon服务是已注册的事件日志发布者后，你可能希望进一步地确定其事件写入位置。为此，可以在命令行中使用`wevtutil gp netlogon`命令，其输出类似于如下格式：

```
name: netlogon
guid: 00000000-0000-0000-0000-000000000000
helpLink: http://go.microsoft.com/fwlink/events.asp?CoName=
Microsoft%20Corporation&ProdName=Microsoft%20Windows%20
%20Operating%20System&ProdVer=6.0.6000.16386&FileName=netmsg.
dll&FileVer=6.0.6000.16386
parameterFileName: %SystemRoot%\System32\kerne132.dll
messageFileName: %SystemRoot%\System32\netmsg.dll
message:
channels:
  channel:
    name: System
    id: 8
    flags: 1
    message:
levels:
opcodes:
```



```
tasks:
keywords:
```

上面的输出中，Channels详细资料中的名称值列出了事件发布者将要写入的日志或多个日志。有些情况下，需要获取所有事件的完整列表（事件发布者向事件日志中写入事件时会用到此列表）。为此，需要将/Ge参数设置为真，如下面实例所示：

```
wevtutil gp Microsoft-Windows-TaskScheduler /ge:true
```

如果事件发布者已经注册了特定的事件，Wevtutil会在标准的发布者配置信息后列出这些事件。另外，你可以指定远程计算机与登录凭据。比如，在下面的实例中，对FileServer25上的事件发布者Microsoft-Windows-TaskScheduler进行了检查：

```
wevtutil gp Microsoft-Windows-TaskScheduler /r:fileserver25
/u:cpandl\williams /p: Cabl@#45898
```

8.1.3 查看与改变日志配置

通过日志配置选项，可以控制事件日志的大小与事件日志的处理方式。默认情况下，事件日志设置为可以设置的最大值。当日志达到最大值时，最早存储的事件会被重写，以防止事件日志的溢出。你也可以将事件日志配置为autobackup与strict retention。激活autobackup时，如果事件日志达到最大值，Windows会自动地将事件日志的副本保存到默认的目录下，并创建新的事件日志以便存储最新事件。激活strict retention时，如果事件日志达到最大值，Windows会丢弃新事件，并生成错误消息宣称事件日志已满。

通过如下的语法格式使用Wevtutil gl命令，可以列出事件日志当前配置信息：

```
wevtutil gl LogName
```

其中，LogName为所要检查的事件日志名。比如，如果要查看应用程序日志的配置，可以在命令行中输入如下命令：

```
wevtutil gl application
```

该命令的输出类似于如下格式：

```
name: application
enabled: true
type: Admin
owningPublisher :
isolation: Application
channelAccess: 0:BAG:SYD:(A;;0xf0007;;;SY)(A;;0x7;;;BA)
logging:
  logFileName: %SystemRoot%\System32\Winevt\Logs\application.evtx
  retention: false
  autoBackup: false
  maxSize: 20971520
publishing:
```

查看日志的配置信息时，主要关注的是日志是否处于激活状态以及默认的日志设置。在上面的实例中，应用程序日志处于激活状态，该文件的全路径为%SystemRoot%\System32\Winevt\Logs\application.evtx，retention处于关闭状态，autobackup处于关闭状态，最大日志规模为20,971,520字节（20,480KB）。

真实场景 isolation属性提供了日志安全性相关的宝贵信息。如果将该属性设置为Application, 则该日志与应用程序日志具备同样的安全许可权限。如果将该属性设置为System, 则该日志与系统日志具备同样的安全许可权限。对这两种情况, 除非通过组策略进行了相应设置, 这两种日志的安全性没有过多的约束。如果将该属性设置为Custom, 则该日志的安全性是自定义的, 只有那些允许访问该日志的组成员才可以访问该日志。比如, 安全日志的isolation属性为Custom, 其相关的安全描述符对可以访问该日志的用户进行了限制, 只有那些被赋予了Manage Auditing And The Security Log这一用户权限的用户才可以访问该日志。默认情况下, 只有管理员才具备这一权限。由于安全日志的这种访问控制机制, 要在命令行中操作本地计算机的安全日志, 就必须使用一个增强的、管理员级的命令提示符。

日志是否处于激活状态是重要的。处于关闭状态的日志是不能使用的, 也无法获取关闭状态日志的详细状态信息。要了解激活状态日志的更多信息, 可以使用Wevtutil gli命令。比如, 如果在命令行中执行命令Wevtutil gli application, 就会列出应用程序日志的状态信息, 其输出信息中包含了类似于如下的信息:

```
creationTime: 2006-06-11T23:39:52.078Z
lastAccessTime: 2006-06-11T23:39:52.078Z
lastWriteTime: 2008-03-28T17:35:14.547Z
fileSize: 20975616
attributes: 32
numberOfLogRecords: 43293
oldestRecordNumber: 27607
```

其中, 应该重点关注的键信息包括如下5个。

- ❑ creationTime。列出了事件日志的创建日期与时间。
- ❑ lastWriteTime。列出了事件写入该日志的最后日期与时间。
- ❑ fileSize。列出了日志当前大小(以字节计数)。
- ❑ numberOfLogRecords。列出了日志中事件的数量。
- ❑ oldestRecordNumber。列出了日志中最早的事件记录编号。

注解 如果最早事件记录编号为1, 则说明Windows尚未重写该日志中的事件。如果最早事件记录编号大于1, 则说明Windows重写了日志中的事件, 重写事件的数量即为最早事件记录的编号。比如, 在上例中, 最早事件记录的编号为27,607, 说明Windows已经重写了该日志中大量事件。

如果需要修改日志的配置, 可以使用Wevtutil sl命令。通过/Ms参数, 可以设置日志最大值(以字节计数)。通过/E参数, 可以激活或关闭日志。/e:true用于激活某日志, /e:false则用于关闭某日志。如果需要确保日志不被重写, 可以使用/r:true参数来激活日志retention。激活后, 如果事件日志达到最大值, Windows会丢弃新事件, 并生成错误消息宣称事件日志已满。但如果同时也激活了autobackup, 情况就会有所不同。如果希望日志达到最大值时自动备份, 可以使用/ab:true参数来激活autobackup。要注意的是, 激活autobackup的同时必须激活retention。

参考如下的一些实例, 有助于了解如何配置事件日志。

将系统日志最大值设置为20,971,520字节(20,480KB):


```
Wevtutil sl System /ms:20971520
Wevtutil sl System /maxsize:20971520
```

关闭Windows RPC debug日志:

```
Wevtutil sl Microsoft-Windows-RPC/Debug /e:false
Wevtutil sl Microsoft-Windows-RPC/Debug /enabled:false
```

激活FileServer86上应用程序日志的retention:

```
Wevtutil sl application /s:FileServer86 /rt:true
Wevtutil sl application /s:FileServer86 /retention:true
```

激活DomainServer18上安全日志的retention与autobackup:

```
Wevtutil sl security /s:DomainServer18 /rt:true /ab:true
Wevtutil sl security /s:DomainServer18 /retention:true
/autobackup:true
```

注解 使用安全日志或远程系统时（或者使用远程系统上的安全日志时），你通常需要使用增强的管理员权限命令提示符对日志进行配置。对远程系统，也可能需要使用替代的登录凭据。

8

8.1.4 导出与操作事件日志

典型情况下，你可能需要为所有关键性的系统保存几个月的日志。但即便设置最大的日志大小，也不能完全满足这一需求，好在还有一些变通的方法。前面讲过，你可以通过Windows定期地对事件日志进行存档，也可以使用命令行或命令行脚本将事件日志导出并保存到特定的位置。

通过Wevtutil epl命令，可以以Windows事件日志格式（.evtx）将事件日志导出到文件中。其基本语法格式为：

```
wevtutil epl LogName SaveLocation
```

其中，*LogName*是待导出的事件日志名，*SaveLocation*是导出的事件日志保存文件的全路径。比如，通过如下命令，可以将应用程序日志导出为C:\Logs\AppLog092908.evtx文件：

```
wevtutil epl Application C:\Logs\AppLog092908.evtx
```

注解 如果创建一个专用的日志存档目录，就可以更方便地定位事件日志。为方便起见，对存档的事件日志进行命名时，应该采用描述性较强的文件名，从文件名本身就可以容易地判断事件日志的类型以及日志存档时间。比如，如果对2009年6月的应用程序日志进行存档，建议使用的较好的存档文件名是AppLogJune2009.evtx。

以这种方式导出事件日志时，Windows会将指定日志的全部内容导出到指定的存档文件，但不会对事件日志进行清除操作。清除事件日志是通过Wevtutil cl命令实现的，这一命令将在8.1.5节讲解。

如果需要导出事件日志中的一部分事件（而不是所有事件），则可以使用保存的XPath查询。在6.4

节中，讨论了XPath查询的基础，以及如何使用XPath查询创建自定义视图。要使用XPath查询导出事件日志中的一部分事件，必须创建包含筛选器（而非自定义视图）的XPath查询。要完成这一任务，最简单的方式是使用事件查看器并遵循如下步骤。

(1) 依次单击“开始”、“管理工具”、“事件查看器”来启动事件查看器。

注解 如果需要处理某台计算机上的事件日志，但没有本地登录该系统，则可以鼠标右键单击事件查看器节点，选择“连接到其他计算机”。之后，在弹出的“选择计算机”对话框的“其他计算机”文本框中，输入主机名、IP地址或目标计算机的完全限定域名。必要的时候，可以使用替代的登录凭据来连接到远程计算机。要建立连接，单击“确定”。

(2) 选择需要筛选与处理的事件日志，在操作菜单，单击“筛选当前日志”。

(3) 在图8-1所示的“筛选当前日志”对话框中，使用记录时间列表选择要包含的事件的时间帧。根据不同的应用需求，你可以选择包含任何时间，前一个小时、过去的12小时、过去的24小时、最后7天、最后30天等事件。

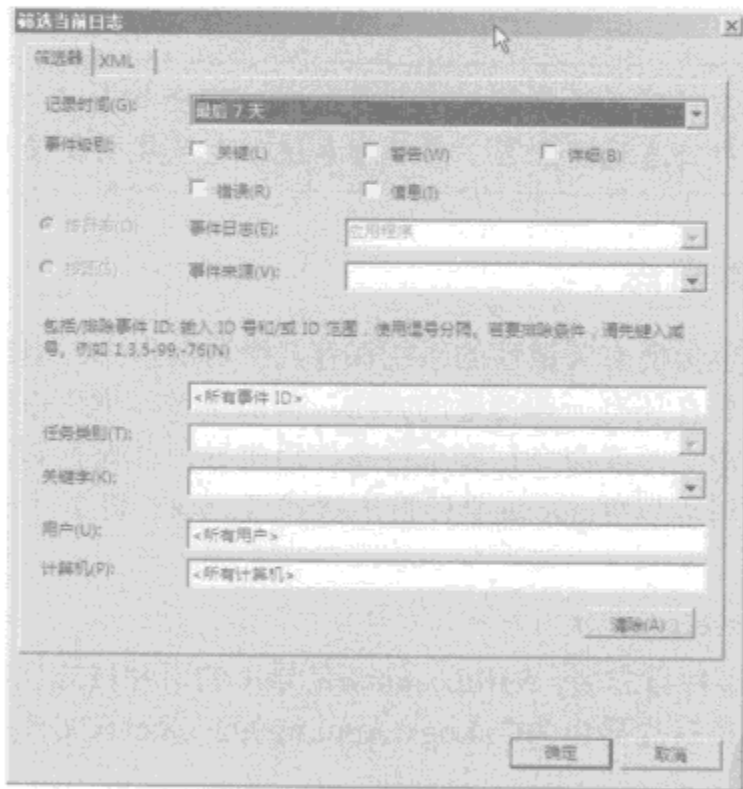


图8-1 创建筛选器，指定要显示的事件类型

(4) 使用事件级别复选框指定要包含的事件级别。选择“详细”，可以获取更多的详细资料。

(5) 你可以为所有事件源创建筛选器，也可以为某特定的事件源子集创建筛选器。要选择需要包含的事件源，可以使用事件源列表。通过选定相应的复选框，可以选择多个事件源。要记住的是，选择某些特定的事件源后，所有其他事件源将被排除在外。

(6) 可选地，使用用户与计算机对话框指定应该包含的用户与计算机。如果没有明确指定待包含的用户与计算机，则所有用户与计算机生成的事件都将默认包含。

(7) 单击XML选项卡来显示相关的XPath查询，如图8-2所示。

(8) 选中查询语句，按Ctrl+A全选，之后按Ctrl+C将查询语句复制到剪贴板。

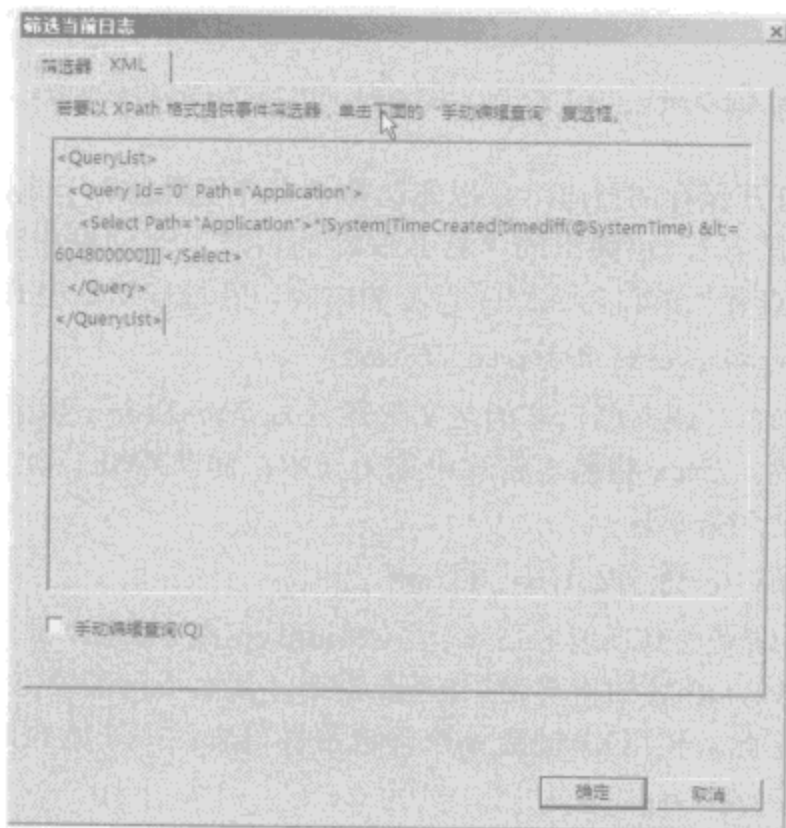


图8-2 查看相关的XPath查询

- (9) 依次单击“开始”、“所有程序”、“附件”、“记事本”打开记事本。
- (10) 按Ctrl+V，将剪贴的内容粘贴到记事本。
- (11) 在记事本的“文件”菜单中，选择“另存为”。在弹出的“另存为”对话框中，选择一个存储位置，在“文件名”文本框中，输入要保存的文件名。另外要注意确保文件扩展名为.xml。
- (12) 在“保存类型”列表中，选择“所有文件(*.*)”，防止记事本将.txt作为文件名的扩展名。
- (13) 单击“保存”，关闭“另存为”对话框。
- (14) 单击“确定”，关闭“过滤当前日志”对话框，并在事件查看器中预览筛选器。

下面给出一个XPath查询示例，其功能是过滤应用程序日志中最近的关键性事件、错误事件与警告事件：

```
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[(Level=1 or Level=2 or Level=3)
and TimeCreated[timediff(@SystemTime) &lt;= 604800000]]]</Select>
  </Query>
</QueryList>
```

通过将/Structuredquery (/Sq) 设置为真，可以使得Wevtutil获知将要使用XPath查询。这种做法不再需要指定待导出的日志，因为筛选器中会对其进行定义，但必须使用如下的语法格式通知Wevtutil将要进行XPath查询：

```
wevtutil epl QueryPath SaveLocation /sq:true
```

其中，*QueryPath*为当前目录中的XPath查询名（或者是XPath查询的全文件路径），*SaveLocation*为待导出文件保存位置的全路径。比如，通过使用如下命令，你可以使用在文件C:\Queries\AppQuery.xml中定义的XPath查询将过滤后的应用程序日志导出为C:\Logs\AppLogFiltered092908.evtx

文件:

```
wevtutil epl C:\Queries\AppQuery.xml C:\Logs\AppLogFiltered092908.evtx
/sq:true
```

在事件查看器中,要打开保存的日志,可以先选择“事件查看器”节点,之后在操作菜单或操作面板中选择“打开保存的日志”,在弹出的“打开保存的日志”对话框中选择要打开的日志。通过Wevtutil,可以使用qe命令查看当前日志或存档日志的内容,其最佳语法格式如下所示:

```
wevtutil qe LogName /c:NumEvents /rd:true /f:text
```

其中,LogName为事件日志名(或存档日志的全文件路径),NumEvents为要读取的事件数量,/rd:true表明要处理的是最近的事件,/f:text将输出格式设置为文本,而非XML。比如,使用如下命令,可以读取应用程序日志中最近的50条事件:

```
wevtutil qe Application /c:50 /rd:true /f:text
```

你也可以使用XPath查询来过滤当前的日志。与wevtutil epl命令类似,通过将/Structuredquery (/Sq)参数设置为真,可以使得Wevtutil获知将要使用XPath查询,从而不再需要指定待导出的日志,因为筛选器中会对其进行定义。据此,使用XPath查询来筛选事件日志的语法格式为:

```
wevtutil qe QueryPath /sq:true
```

其中,QueryPath为当前目录中的XPath查询名,或者是XPath查询的全文件路径,比如:

```
wevtutil qe C:\Queries\AppQuery.xml /sq:true
```

你可以使用/C、/Rd、/F等参数来对输出信息进行约束,前面已经讨论,下面再给出一个实例:

```
wevtutil qe C:\Queries\AppQuery.xml /sq:true /c:50 /rd:true /f:text
```

你也可以使用/R、/U、/P等参数来查询远程计算机上的事件,前面已经讨论,下面再给出一个实例:

```
wevtutil qe C:\Queries\AppQuery.xml /sq:true /c:50 /rd:true /f:text/
r:PrintServer23 /u:adatum\williams /p:Fiber
```

8.1.5 清除事件日志

事件日志已满时,就需要对其进行清除。要完成这一任务,可以以如下的语法格式使用Wevtutil cl命令:

```
wevtutil cl LogName
```

其中,LogName为待清除的事件日志名,比如:

```
wevtutil cl Application
```

在清除日志文件之前,可能需要对其进行备份,以便保存事件日志内容的副本。通过使用/Backup (/Bu)参数,其后跟随备份文件名或文件路径,可以规定在清除事件日志前对其进行备份并将其保存到指定的位置。需要注意的是,必须包含文件扩展.evtx,如下面实例所示:

```
wevtutil cl Application /bu:C:\Logs\AppLogFiltered.evtx
```

8.2 企业级集中化事件记录机制

在第6章与本章,已经讲解了可用于事件日志处理的很多技术。尽管你可以创建命令行脚本来检

查多台计算机上的事件日志，并将这些信息复制到一个中央存储位置以便查阅，但这不是唯一的方法。通过配置事件转发机制，可以实现集中化事件记录。通过事件转发，可以将所有事件或特定类型事件转发到指定的、用于集中化事件记录的计算机。比如，你可能需要将组织内所有工作站与服务器配置为，将安全日志中的认证失败事件转发到用于集中化事件记录的计算机，以便对事件进行综合分析，并实时地检测出入侵企图。再如，对关键性的服务器，你可能需要将应用程序日志与系统日志中的关键性事件与错误事件转发到用于集中化事件记录的计算机，之后对事件进行综合分析，并实时地检测出应用程序错误与其他需要管理员注意的问题。配置集中化的事件记录机制是一个多步骤的过程。首先，要配置事件转发与事件收集机制。之后，在指定的、用于集中化事件记录的计算机上，必须创建订阅，其中指定要转发事件的类型与转发事件的日志源（以每台源计算机为基础）。在用于集中化事件记录的计算机上，默认情况下，转发的事件被收集到转发事件日志中。

8.2.1 配置事件转发与收集

事件转发的体系结构是非常灵活的。比如，你可以将用于集中化事件记录的计算机配置为向其他用于集中化事件记录的计算机转发事件。由于转发协议（WS-Management）在底层上实际使用了超文本传输协议（HTTP）与安全HTTP协议（HTTPS），因此，只要相关的TCP端口是开放的，也可以使得转发事件顺利地通过防火墙。在标准的配置中，这意味着，如果使用HTTP协议，则TCP 80端口必须是开放的；如果使用HTTPS协议，则TCP 443端口必须是开放的。

要使用事件转发，需要配置与激活相关计算机上的事件转发功能，并在用于集中化事件记录的计算机或计算机上为转发事件创建订阅。在域中，通过如下步骤，可以为转发事件配置事件转发与收集机制。

(1) 要在运行Windows Vista或Windows Server 2008的计算机上配置事件转发功能，必须登录所有源计算机，并完成如下任务。

- ☐ 将Windows Remote Management Service (WinRM) 服务的启动方式设置为自动（延迟的启动）。
- ☐ 启动WinRM服务。
- ☐ 创建WinRM侦听程序。
- ☐ 将WinRM服务设置为防火墙例外（如果防火墙处于打开状态）。

这些任务并不需要通过不同的操作分别完成，实际上，通过在一个增强的命令提示符中键入**winrm quickconfig**命令，就可以完成上面的配置任务，并在HTTP://*上创建一个WinRM侦听程序，以便在源计算机上接收到任意IP地址的WS-Management请求。执行该命令后，会弹出确认信息，选择“是”。

(2) 要配置事件收集，登录用于集中化事件记录的计算机，在一个增强的命令提示符中，键入**wecutil qc**命令。该命令将启动Windows Event Collector服务，并将该服务的启动方式设置为手动启动。弹出确认信息后，选择“是”。

(3) 在活动目录中，为事件收集计算机创建一个全局组，并将该计算机添加到该组中，在第15章15.2.1与15.2.2中，会对这一问题进行讨论。

(4) 在每台源计算机上，将事件收集计算机的全局组加入到本地管理员组。为完成这一任务，可以在一个增强的命令提示符中使用**net localgroup**命令。其语法格式为：**net localgroup Administrators CollectorGroup /add**，其中CollectorGroup是为事件收集计算机创建的全局组名，比如**net localgroup Administrators Collectors /add**。

8.2.2 创建订阅

要创建订阅，最简单的方法是使用事件查看器，并遵循如下步骤。

(1) 依次单击“开始”、“管理工具”、“事件查看器”来启动事件查看器。如果尚未登录用于集中化事件收集的计算机，鼠标右击事件查看器节点，选择“连接到其他计算机”，在弹出的“选择计算机”对话框的“其他计算机”文本框中，输入主机名、IP地址或目标计算机的完全限定域名。必要的时候，可以使用替代的登录凭据来连接到远程计算机。要建立连接，单击“确定”。

(2) 鼠标右击订阅节点，选择“创建订阅”。

(3) 在图8-3所示的“订阅属性”对话框中，为订阅键入名称，比如“所有文件服务器”，订阅名被设置为订阅标识符。可选地，还可以为其输入相应的描述信息。

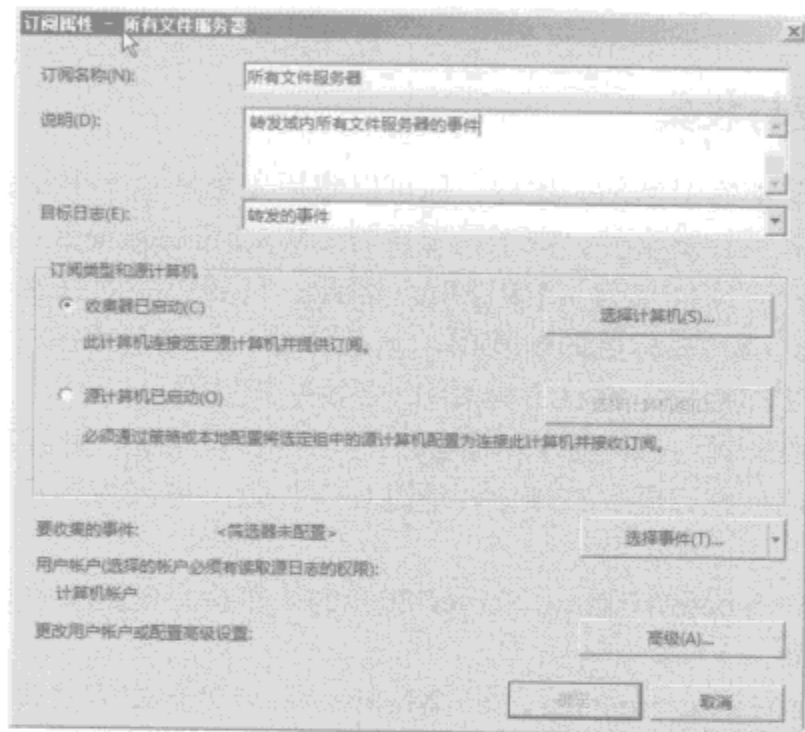


图8-3 创建用于转发事件的订阅

(4) 默认情况下，转发事件日志会被选择为目标日志。如果需要将事件转发到专门用于事件收集的计算机，就可能需要将事件转发到其他目标日志。要注意的是，这样做时，应该确保所创建的查询只对该特定日志有效，以便降低其他可能的混淆。

(5) 事件收集计算机发起的事件转发是最容易配置的，也是缺省的设置。要指定将事件转发到服务器的计算机，单击“选择计算机”，在弹出的“计算机”对话框中，单击“添加域计算机”。在“选择计算机”对话框中，键入转发事件计算机的账号名，之后单击两次“确定”。如果需要，重复这一过程。

(6) 单击“选择事件”，这将弹出“查询筛选器”对话框。

(7) 使用事件级别检查复选框指定要包含的事件级别。选择“详细”，会获取更多的详细资料。

(8) 大多数情况下，你可能需要为单一日志或多个日志创建一个筛选器。为此，要确保“按日志”被选定，之后使用事件日志列表选择要过滤的单一或多个日志。

(9) 如果需要针对特定的事件源过滤选定的一个或多个日志，而不是针对所有事件源，则需要使用事件源列表来选择要包含的事件源。通过选定相应的复选框，可以选择多个事件源。要记住的是，选择某些特定的事件源后，所有其他事件源将被排除在外。另外，注意不要选择“按源”选项，因为

这样会清除所做的日志选定。

(10) 默认情况下, Windows Event Collector服务使用集中化事件收集计算机的机器账号来读取源日志。要使用其他登录凭据, 可以单击“高级”。在弹出的“高级订阅设置”对话框中, 选择“特定用户”, 单击“用户和密码”按钮。在“订阅源的凭据”对话框中, 输入要使用账号的用户名与密码, 单击两次“确定”。你可以以域\用户的格式来指定用户的登录域, 比如cpandl\williams。

(11) 单击“确定”创建订阅, 在创建了订阅并且选定的计算机开始转发事件后, 就可以在目标日志中查看转发的事件。

上面讲解了如何使用事件查看器创建订阅及其具体步骤, 下面讲述如何在命令行中完成这些复杂的任务。通过使用表8-3中的Wecutil命令, 可以在集中化事件收集计算机上对订阅进行创建与管理。与Weventutil命令一样, Wecutil命令也有完整的长格式与缩略的短格式。

表8-3 可用的Wecutil命令

长 格 式	短 格 式	描 述
enum-subscription	Es	列出计算机上现存的订阅
get-subscription	Gs	显示订阅配置的详细资料
get-subscriptionruntimestatus	Gr	显示订阅运行时状态
set-subscription	Ss	修改订阅配置
create-subscription	Cs	创建新订阅
delete-subscription	Ds	删除订阅
retry-subscription	Rs	重试某订阅的所有源
quick-config	Qc	激活并配置Windows Event Collector服务

要创建订阅, 可以使用Wecutil cs命令与XML配置文件。配置文件中包含了XPath查询, XPath查询定义了要转发的事件与源计算机, 并为订阅分配了订阅标识符。使用事件查看器创建订阅时, 事件收集计算机发起的事件转发是最容易配置的, 也是默认的设置。

通过在命令提示符中使用wecutil cs/?命令, 可以显示示例XML配置文件(可以将其复制与粘贴到文本编辑器)。XML配置文件具有类似于如下的格式:

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/
subscription">
  <Uri>http://schemas.microsoft.com/wbem/wsmn/1/windows/EventLog</Uri>
  <!-- Use Normal (default), Custom, MinLatency, MinBandwidth -->
  <ConfigurationMode>Normal</ConfigurationMode>
  <Description>Forward Sample Subscription</Description>
  <SubscriptionId>SampleSubscription</SubscriptionId>
  <Query><![CDATA[
    <QueryList>
      <Query Path="Application">
        <Select>*</Select>
      </Query>
    </QueryList>
  ]]></Query>
  <EventSources>
    <EventSource Enabled="true">
```

```

    <Address>mySource.myDomain.com</Address>
    <UserName>myUserName</UserName>
    <Password>*</Password>
  </EventSource>
</EventSources>
<CredentialsType>Default</CredentialsType>
<Locale Language="EN-US"></Locale>
</Subscription>

```

在把XML配置文件复制并粘贴到文本编辑器中之后，就可以对其进行编辑。要编辑的关键字段包括描述、订阅标识符、XPath查询以及事件源。

描述部分提供了订阅的一些通常的描述信息，有助于管理员了解订阅如何使用。描述信息是使用Description元素定义的，必要的时候，可以对<Description>与</Description>标签中的描述文本进行修改。下面给出一个实例：

```
<Description>Forwards important events from file servers</Description>
```

订阅标识符用于唯一地标识集中化事件收集计算机上的订阅，使用SubscriptionId元素进行描述。必要的时候，可以对<SubscriptionId>与</SubscriptionId>标签中的标识符进行修改，下面给出一个实例：

```
<SubscriptionId>All File Servers</SubscriptionId>
```

元素Query之间的QueryList元素定义了XPath查询，你可以向其中复制与粘贴任意现存的XPath查询（定义了事件日志筛选器）。不过，由于事件转发是实时进行的，你可能需要移除XPath查询中根据事件创建时间进行过滤的部分。

与使用Weventutil处理单条事件日志不同，通常希望创建单一的XPath查询来处理多个日志，因为这会降低需要创建的订阅数量。下面给出一个XPath查询实例，对应用程序日志，系统日志，DNS Server日志中的关键性事件、错误事件与警告事件进行过滤：

```

<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[(Level=1 or Level=2 or Level=3)]]
  </Select>
  <Select Path="System">*[System[(Level=1 or Level=2 or Level=3)]]
</Select>
  <Select Path="DNS Server">*[System[(Level=1 or Level=2 or Level=3)]]
</Select>
</Query>
</QueryList>

```

元素EventSources之间的单独EventSource元素用于定义不同的源计算机。每台将要转发事件（基于此前定义的XPath查询）的计算机必须由自己的EventSource元素进行描述，包括该计算机的完全限定域名或IP地址。下面的实例中，激活了Cpandl.com域内的FileServer24、FileServer26、FileServer28等计算机的事件转发功能：

```

<EventSources>
  <EventSource Enabled="true">
    <Address>fileseryer24.cpandl.com</Address>
  </EventSource>
  <EventSource Enabled="true">
    <Address>fileseryer26.cpandl.com</Address>
  </EventSource>

```

```
<EventSource Enabled="true">
  <Address>fileseryer28.cpandl.com</Address>
</EventSource>
</EventSources>
```

默认情况下，Windows Event Collector服务使用集中化事件收集计算机的机器账号来读取源日志。要使用其他登录凭据，必须提供必要的登录信息。为此，一种做法是将其作为EventSource元素的一部分，如下面实例所示：

```
<EventSource Enabled="true">
  <Address>fileseryer24.cpandl.com</Address>
  <UserName>williams</UserName>
  <Password>* </Password>
</EventSource>
```

上面实例中，指定了为计算机FileServer24使用的登录凭据。由于口令指定为*，在创建订阅时，会遇到提示信息要求输入登录凭据，所提供的登录凭据会与该订阅进行关联并安全存储。

注解 在配置文件中定义其他登录凭据时，应该总是使用*作为口令，而不直接输入实际的口令。在配置文件中直接输入实际的口令是一种安全性较差的做法。

上面的示例配置文件中没有出现，但实际上被赋予默认值的元素包括下面6个。

- **ConfigurationMode**。将事件传递优化方法设置为正常、最小化带宽或最小化滞后时间，包含在<ConfigurationMode>标签与</ConfigurationMode>标签之间。
- **DeliveryMode**。将传输方法设置为Push模式或Pull模式（默认情况下是Pull），包含在begin<DeliveryMode>标签与end</DeliveryMode>标签之间。
- **ReadExistingEvents**。用于指定是读取现存的事件还是只读取新事件，包含在begin<ReadExistingEvents>标签与end</ReadExistingEvents>标签之间。默认情况下，该元素取值为False，表示只读取新事件。
- **LogFile**。用于指定事件将要转发到的目标日志，包含在begin</LogFile>标签与end</LogFile>标签之间。指定日志名时，注意要去除日志名中的空格。比如，要指定日志Forwarded Events，就应该将引用为ForwardedEvents。
- **TransportName**。用于指定使用的传输协议，或者为HTTP，或者为HTTPS，包含在begin<TransportName>标签与end</TransportName>标签之间。如果指定HTTPS作为传输协议，必须同时在源计算机上为HTTPS配置一个WinRM侦听程序。
- **TransportPort**。指定使用的传输端口，包含在begin<TransportPort>标签与end</TransportPort>标签之间。要注意的是，这里指定的端口应该与源计算机上配置WinRM侦听程序时使用的端口相同。典型情况下，HTTP协议对应80端口，HTTPS协议对应443端口。

示例8-1展示了一个用于Wecutil的配置文件中，其中使用了上面讨论的元素与选项。

示例8-1 用于Wecutil的配置文件中

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/
events/subscription">
  <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
  <!-- Use Normal (default), Custom, MinLatency, MinBandwidth -->
```

```

<ConfigurationMode>Normal</ConfigurationMode>
<Description>Forwards events from file servers in the domain</
Description>
<SubscriptionId>All Servers</SubscriptionId>
<DeliveryMode>Pull</DeliveryMode>
<ReadExistingEvents>False</ReadExistingEvents>
<LogFile>ForwardedEvents</LogFile>
<TransportName>HTTPS</TransportName>
<TransportPort>443</TransportPort>
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[(Level=1 or Level=2 or
Level=3)]]</Select>
    <Select Path="System">*[System[(Level=1 or Level=2 or Level=
3)]]</Select>
    <Select Path="DNS Server">*[System[(Level=1 or Level=2 or Level
=3)]]</Select>
  </Query>
</QueryList>
<EventSources>
  <EventSource Enabled="true">
    <Address>fileserver24.cpandl.com</Address>
  </EventSource>
  <EventSource Enabled="true">
    <Address>fileserver26.cpandl.com</Address>
  </EventSource>
  <EventSource Enabled="true">
    <Address>fileserver28.cpandl.com</Address>
  </EventSource>
</EventSources>
<CredentialsType>Default</CredentialsType>
<Locale Language="EN-US"></Locale>
</Subscription>

```

创建了XML配置文件之后，就可以以如下的语法格式使用Wecutil cs命令来创建订阅：

```
wecutil cs ConfigFile
```

其中，*ConfigFile*为当前目录中的配置文件名，或其他目录中配置文件的全文件路径。比如，如果需要文件C:\Evtforwarding\config1.xml来创建订阅，可以使用如下命令：

```
wecutil cs c:\evtforwarding\config1.xml
```

创建订阅时，可以指定共享的登录凭据，而不是必须使用/Cun与/Cup参数在配置文件中指定的登录凭据。/Cun参数用于设置用户名，/Cup参数用于设置口令，如下面实例所示：

```
wecutil cs c:\evtforwarding\config1.xml /cun:cpandl\williams /cup:Rover
```

8.2.3 管理订阅

创建了订阅并且选定的计算机开始转发事件之后，就可以在目标日志中看到转发的事件。在事件查看器中选择订阅节点，就可以看到所创建的订阅。根据名称、状态、类型、源计算机数量、目标日志、描述信息等不同属性，事件查看器列出了每一个订阅。单击并选中某个订阅之后，就可以使用操作面板或操作菜单中的如下一些选项对其进行管理。

- **属性。**显示订阅属性对话框，通过该对话框，可以对订阅进行修改。可用的选项与前面讨论过的类似，但不能修改订阅名或类型。
- **禁用。**禁用该订阅，使得事件不再被转发或收集。选定后，订阅状态将改变为禁用。
- **激活。**激活该订阅，使得事件被转发或收集。选定后，订阅状态将改变为激活。
- **运行时状态。**显示订阅的运行时状态，包括该订阅是激活的还是禁用的，以及源计算机上是否出现了事件收集错误等信息。如果源计算机上出现了错误状态，就可以选定该条目查看进一步的详细资料。选定了该源计算机条目后，还可以选择禁用或激活该源计算机的事件转发功能。
- **删除。**永久性删除某订阅。

Wecutil提供了用于完成这些任务的命令。要根据名称列出某计算机上所有订阅，可以在命令提示符中使用**wecutil es**命令。要列出某订阅配置详细资料，可以使用**wecutil gs**命令，其后跟随订阅名。如果订阅名中包含空格，则需要使用引号对其进行封装，比如：

```
wecutil gs "All File Servers"
```

输出信息如下所示，展示了该订阅的配置详细资料：

```
Subscription Id: all servers
SubscriptionType: CollectorInitiated
Description: all servers in the domain
Enabled: false
Uri: http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog
ConfigurationMode: Normal
DeliveryMode: Pull
DeliveryMaxLatencyTime: 900000
HeartbeatInterval: 900000
Query: <QueryList><Query Id="0"><Select Path="Application">* [System[(Level=1
    or
    Level=2 or Level=3)]]</Select><Select Path="System">* [System[(Level=1
    or Level=2
    or Level=3)]]</Select><Select Path="DNS Server">* [System[(Level=1 or
    Level=2
    or Level=3)]]</Select></Query></QueryList>
ReadExistingEvents: false
TransportName: HTTP
TransportPort: 80
ContentFormat: RenderedText
Locale: en-US
LogFile: Microsoft-Windows-DateTimeControlPanel/Operational
PublisherName :
CredentialsType: Default

EventSource[0] :
    Address: MAILSERVER84.cpandl.com
    Enabled: true
EventSource[1] :
    Address: ROOM5.cpandl.com
    Enabled: true
```

你可以使用**Wecutil ss**命令来修改订阅的设置。使用该命令时，你可以使用可用的参数来修改订阅

的配置。然而，要修改订阅，最简单的方法是创建一个新的配置文件，或直接在原始的配置文件中进行修改，之后以如下语法格式运行Wecutil ss命令：

```
wecutil ss SubscriptionName ConfigFile
```

其中，*SubscriptionName*为订阅名，*ConfigFile*是当前目录中的配置文件名，或者其他目录中配置文件的全文件路径。比如，如果需要使用文件C:\Evtforwarding\config1.xml来创建订阅，可以使用如下命令：

```
wecutil ss "All File Servers" c:\evtforwarding\config1.xml
```

创建订阅时，可以指定共享的登录凭据，而不是必须使用/Cun与/Cup参数在配置文件中指定的登录凭据。/Cun参数用于设置用户名，/Cup参数用于设置口令，如下面实例所示：

```
wecutil ss "All File Servers" c:\evtforwarding\config1.xml  
/cun: cpandl\williams /cup:Rover
```

你也可以使用Wecutil ss命令来激活或禁用订阅，这分别是通过将/E参数设置为True与False实现的。比如，下面实例中，禁用了All File Servers这一订阅：

```
wecutil ss "All File Servers" /e:false
```

要获取某订阅的运行状态，可以在命令提示符中输入wecutil gr命令，其后跟随订阅名。其输出信息提供了近期错误信息与每台源计算机上事件转发处于激活或禁用状态的详细资料，下面给出一个实例：

```
Subscription: all servers  
RunTimeStatus: Disabled  
LastError: 0  
EventSources :  
    MAILSERVER84.cpandl.com  
        RunTimeStatus: Disabled  
        LastError: 0  
    ROOM5.cpandl.com  
        RunTimeStatus: Disabled  
        LastError: 0
```

如果不再需要使用某订阅，就可以使用Wecutil ds命令对其进行永久性删除。在命令行中，输入wecutil ds，其后跟随要删除的订阅名，比如wecutil ds “all file servers”。

8.3 性能日志

Windows Vista与Windows Server 2008中使用了数据收集器集与报告。通过数据收集器集，可以指定需要追踪的性能对象集与计数器。创建了一个数据收集器集后，可以很方便地启动或终止对其所包含的性能对象与计数器的监控。在某种程度上，这使得数据收集器集类似于早期Windows版本中的性能日志。然而，实际上数据收集器集要比早期的性能日志复杂与灵活得多。

8.3.1 开始使用数据收集器集

你可以创建并使用不同类型的数据收集器集，其类型包括下面5个。

- 警报。此类型用于在特定事件发生或特定性能指标达到时通知用户。
- API。此类型用于那些在与其相关的源提供者事件发生时记录数据的数据收集器。

- **配置**。此类型用于那些会记录特定注册表路径改变的数据收集器。
- **性能计数器**。此类型用于那些在预定义时间间隔达到后在选定的计数器上记录数据的数据收集器。
- **跟踪**。此类型用于那些在相关事件发生时记录性能数据的数据收集器。

Windows Vista与Windows Server 2008使用事件踪迹来追踪大量不同类型的性能统计数据。有些事件跟踪在配置上与操作系统一起自动启动，这些事件跟踪称为启动事件跟踪。大多数时候，使用最广泛的两种类型数据收集器是性能计数器与警报，也是本节将要讲述的。

在图形用户界面中，可以在可靠性和性能监视器中创建与管理数据收集器。在数据收集器集节点下，可以看到用户自定义数据收集器节点、系统定义的数据收集器节点、事件跟踪会话以及启动事件跟踪会话等节点。通过扩展这些节点，就可以查看事件收集器与跟踪的相关条目。

在命令行中，你可以使用Logman命令来操作事件收集器。大多数情况下，这一命令需要一个增强的、管理员权限的命令提示符。通过键入logman或logman query命令，可以查看计算机上当前配置的数据收集器。成功执行后，会输出一个列表，其中包含了数据收集器的名称、类型与状态，如下面实例所示：

Data Collector Set	Type	Status
High CPU	Alert	Started
Memory Usage	Alert	Stopped

通过使用如下的语法格式，也可以在远程计算机上运行该命令：

```
logman query -s RemoteComputer
```

其中，*RemoteComputer*为待检查的远程计算机的主机名或IP地址。由于是否具备对该远程计算机的访问权限取决于当前用户登录凭据，因此，在使用该命令前，要确保是以具备对远程计算机适当访问权限的账号登录的。大多数情况下，还需要在增强的命令提示符中运行该命令。

要查看特定数据收集器的详细资料，可以输入logman query命令，其后跟随该数据收集器的名称。输出信息是一个详尽的列表，类似于如下的格式：

```
Name :          cpu
Status :        Stopped
Root Path :     %systemdrive%\PerfLogs\Admin\CPU
Segment:        Off
Schedules :     On
Run as :        SYSTEM
```

```
Name:           cpu\DataCollector01
Type :          Alert
Sample Interval : 15 second(s)
Event Log:      Off
```

```
Thresholds:
  \Processor(_Total)\% Processor Time>98
```

8.3.2 操作数据收集器集

使用可靠性和性能监视器时，通过鼠标右击某数据收集器，之后选择开始或停止，就可以开始或停止对该数据收集器的记录。在命令行中，可以使用Logman Start命令与Logman Stop命令来分别完成

同样的任务。启动一个数据收集器的命令语法格式为：

```
logman start CollectorName
```

其中，*CollectorName*为要启动的数据收集器的名称，比如：

```
logman start "General Activity Monitor"
```

终止一个数据收集器的命令语法格式为：

```
logman stop CollectorName
```

其中，*CollectorName*为要终止的数据收集器的名称，比如：

```
logman stop "General Activity Monitor"
```

在可靠性和性能监视器中，通过鼠标右击某数据收集器，之后选择保存模板，可以将某数据收集器保存为模板，并作为其他数据收集器的参照与基础。在另存为对话框中，选择一个目录，为该模板键入名称，之后单击“保存”。数据收集器模板是以XML文件格式保存的，可以复制到其他系统，也可以在创建新的数据收集器时使用。

如果需要在命令行中将数据收集器设置导出到XML文件，可以以如下语法格式使用Logman export命令：

```
logman export CollectorName -xml OutputFile
```

其中，*CollectorName*是要使用的数据收集器名，*OutputFile*是数据收集器设置信息将要写入的XML文件名，比如：

```
logman export "General Activity Monitor" -xml GeneralCollectorConfig.xml
```

通过Logman import命令，就可以使用模板来导入并创建（或重建）数据收集器。其语法格式与导出命令类似，如下所示：

```
logman import CollectorName -xml InputFile
```

其中，*CollectorName*是要使用的数据收集器名，*InputFile*是要读出数据收集器设置信息的XML文件名。使用-S参数，可以指定远程计算机。使用-U参数，可以指定要使用哪个用户的登录凭据。指定用户时，可以在用户名后跟随口令，也可以不输入口令或用*代替口令。这样会在具体使用时由系统给出提示信息要求输入口令，比前一种方式更具安全性。下面的实例中，在远程计算机（FileServer86）上使用用户账号WilliamS导入数据收集器：

```
logman import "General Activity Monitor" -xml GeneralCollectorConfig.xml  
-s FileServer86 -u Williams
```

在可靠性和性能监视器中，如果需要删除用户定义的数据收集器，可以通过鼠标右击该数据收集器，之后选择“删除”即可。如果数据收集器处于运行状态，则需要先终止其运行，之后对其进行删除。删除收集器的同时也删除了与其相关的报告。

在命令行中，可以使用Logman delete命令删除用户定义的数据收集器，其语法格式如下：

```
logman delete CollectorName
```

其中，*CollectorName*为要删除的数据收集器的名称，比如：

```
logman delete "General Activity Monitor"
```

对运行状态的数据收集器，在删除之前必须先终止它。

通过使用-S参数，可以删除远程计算机上的数据收集器。必要的时候，还可以使用-U参数指定可替代的登录凭据，其语法格式为：

`-u UserName Password`

其中，*UserName*是用于访问远程计算机的用户账号名，*Password*是该账号的口令（可选的）。

8.3.3 收集性能计数器数据

在特定的示例时间间隔，你可以使用数据收集器来记录选定的计数器的性能数据。比如，可以每隔15分钟对CPU的性能数据进行一次采样。默认的记录位置是%SystemDrive%\PerfLogs\Admin，日志文件大小可以非常快速地增长。如果要长时间地记录某些性能数据，最好是把日志文件保存在空余存储空间较大的驱动器上。要记住的是，日志文件的更新越频繁，所占据的驱动器空间就越大，系统中CPU资源使用就越多。

要收集性能计数器数据，需要遵循如下步骤。

(1) 在可靠性和性能监视器中的数据收集器集节点下，鼠标右击左面板的“用户定义”节点，单击“新建”，之后选择“数据收集器集”。

(2) 在“创建新的数据收集器集”向导中，键入数据收集器名，比如**General Activity Monitor**。之后选择“手动创建（高级）”选项，单击“下一步”。

(3) 在“创建新的数据收集器集”页面，“创建数据日志”选项是默认选定的，选择“性能计数器”复选框，单击“下一步”。

(4) 在“创建新的数据收集器集”页面，单击“添加”。在弹出的“可用计数器”对话框中，选择要追踪的性能计数器，选取完毕后，单击“确定”。

(5) 在图8-4所示的“创建新的数据收集器集”页面中，键入示例间隔，并选择秒、分、小时、天、星期等时间计数单位。比如，如果选择每隔30秒进行一次采样，则事件日志每隔30秒进行一次更新。选取完毕后，单击“下一步”。

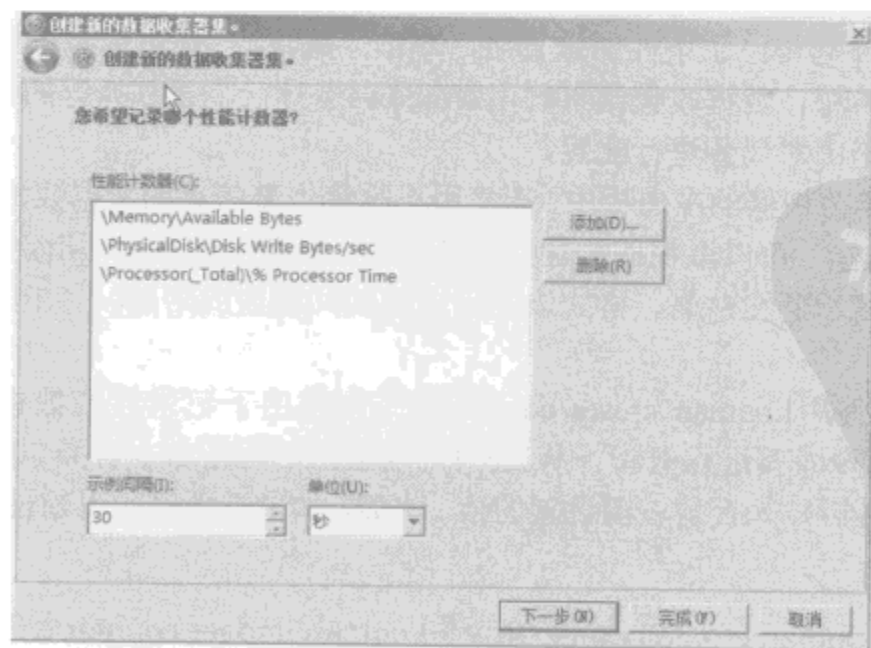


图8-4 设置示例间隔

(6) 在“你希望将数据保存在什么位置？”页面，键入用于记录收集数据的文件根路径。也可以

单击“浏览”，之后使用“浏览到文件夹”对话框来选择日志目录。完毕后，单击“下一步”。

(7) 在“创建新的数据收集器集”页面，“身份”对话框将<默认>作为用户，这意味着日志将以默认系统账号的特权与权限运行。如果需要以其他用户的特权与权限运行日志，单击“更改”，键入指定账号的用户名与密码，之后单击“确定”。用户名可以以域\用户格式输入，比如adatum\williams代表Adatum域内的WilliamS账号。

(8) 选择“保存并关闭”选项，单击“完成”。系统将保存数据收集器集，关闭向导，之后打开相关的属性对话框。

(9) 默认情况下，事件收集是手工启动的。要对事件收集方式进行配置，单击“计划”选项卡，单击“添加”。之后可以设置活动的范围、开始时间以及事件收集的运行天数，如图8-5所示。

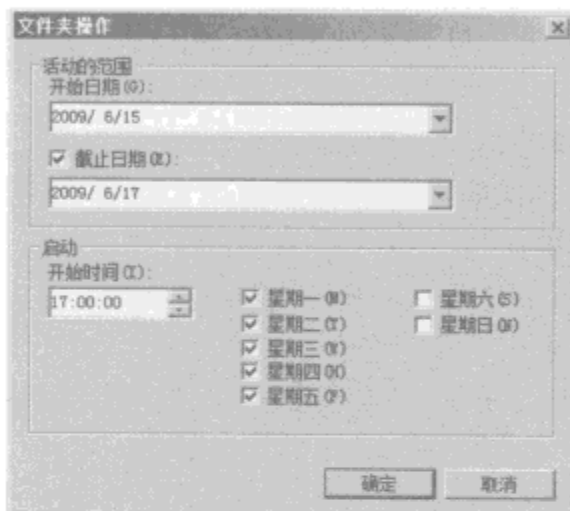


图8-5 为性能记录配置运行计划

(10) 默认情况下，只有在计划中设置了截止日期时，才会在条件达到时停止收集。使用“停止条件”选项卡中的选项，可以将事件收集配置为在经过指定的时间间隔（比如7天，或者在事件日志达到设定的最大值）后手工终止。

(11) 完成了对收集计划与停止条件的设置后，单击“确定”。你可以按照8.3.1节中的讲解对数据收集器进行管理。如果希望在数据收集结束时由Windows运行一个计划执行的任务，可以在“属性”对话框的“任务”选项卡中对任务进行配置。

(12) 在可靠性和性能监视器的左面板中，选择用于收集告警信息的数据收集器，之后选择“属性”。在“性能计数器”选项卡上，可以添加或移除性能计数器、设置示例间隔、指定最大示例数等。在“文件”选项卡上，可以设置示例文件名、格式以及日志模式（覆盖或附加）。完毕后，单击“确定”，保存所做设置。

在命令行中，可以使用Logman create counter命令创建用于记录性能数据的数据收集器，使用Logman update counter命令对其进行更新。从表8-4可以看出，两条命令有类似的可用参数集。对很多参数，可以使用多余的破折号来否定以前设置的值。比如，如果需要停用以前使用-U指定的登录凭据，就可以使用-U。

表8-4 Logman create counter命令与Logman update counter命令的参数

参 数	描 述	可否逆选
-a	将输出附加到现存的日志文件	是

(续)

参 数	描 述	可否逆选
-b<mm/dd/yyyy h:mm:ss [AM PM]>	对数据收集器进行调度, 使其在特定的时间开始	
-c <path [path[...]]>	标识要收集的性能计数器 (在未使用计数器文件时)	
-cf <filename>	标识计数器文件, 其中包含了要收集的性能计数器 (每个计数器一行)	
-cnf <[[hh:]mm:]ss>	当指定的时间间隔已到, 或日志文件达到最大值时, 创建一个新文件	是
-config <filename>	创建设置文件, 其中包含了命令选项	
-e<mm/dd/yyyy h:mm:ss [AM PM]>	对数据收集器进行调度, 使其在指定的时间终止	
-f <bin bincirc csv tsv sql>	为日志文件设置输出格式	
-m <[start] [stop] [[start] [stop][...]]>	将自动启动或终止改变为手工启动或终止	
-max <value>	以MB为计数单位设置日志文件最大值, 或SQL日志中记录数	是
-o <path dsn!log>	指定输出日志文件的路径, 或SQL数据库中的DSN与日志集名	
-ow	重写现存的日志文件	是
-r	每天会在指定的时间段内重复执行数据收集器	是
-rc <command>	每次日志关闭时运行指定的命令	是
-rf <[[hh:]mm:]ss>	为数据收集器设定运行持续时间	
-s <computer>	在指定的远程计算机上执行命令	
-sc <value>	设定要收集的示例的最大值	
-si <[[hh:]mm:]ss>	设定性能计数器的示例间隔	
-u <user> [<password>]	为远程计算机设置登录用户名以及口令 (可选的)。如果没有设定口令, 在实际应用中会遇到要求输入口令的提示信息	是
-v <nnnnnn mmddhhmm>	为日志名结尾添加版本信息	是
-y	对所有提示信息应答“是”, 从而不会再遇到提示信息	

使用Logman create counter命令为性能计数器创建数据收集器的最简单的方式是使用如下的语法格式:

```
logman create counter DataCollectorName -c Counter
```

其中, DataCollectorName是为数据收集器设置的唯一名称, Counter设置了相对计数器路径, 相对计数器路径是与6.5.2节所讨论的绝对计数器路径不同的。在相对计数器路径中, 并没有具体指定是哪一台计算机, 这也是称其为相对路径的原因。

如第7章中所阐述的, 性能计数器使用相对路径, 其语法格式为:

```
\ObjectName\ObjectCounter
```

其中, ObjectName为计数器对象名, ObjectCounter为操作的对象计数器。为此, 如果需要创建一个数据收集器来追踪可用的内存计数器, 可以使用如下命令:

```
logman create counter MemCounterCollector -c "\memory\available mbytes"
```

上面的命令创建了数据收集器, 但Logman并不启动计数器。你或者可以像前面讨论的手工启动或

终止计数器，或者对其进行调度，使其在当前日期与时间之后的某个时间段内运行。下面的实例中，在2009年6月15日5:30 P.M启动该计数器，直至2009年6月17日9:30 P.M终止该计数器：

```
logman create counter MemCounterCollector -c "\memory\available mbytes"
-b 06/15/2009 05:30PM -e 06/17/2009 09:30PM
```

如果设置了错误的启动时间与终止时间，可以使用Logman counter update命令来修改数据收集器的设置。如果只设置了启动时间而没有设置终止时间，则该数据收集器会一直运行下去。如果设置了终止时间，则必须设置启动时间。要注意的是，如果启动时间在设置上早于当前时间，则数据收集器不会自动启动。

如果没有设置示例间隔或示例的最大值，则数据收集器会每隔15秒进行一次采样，并且一直持续进行。通过-Si参数，可以设置示例间隔，-Sc参数则用于对示例数量进行计数。下面的实例创建了一个新的计数器，并将示例间隔设置为10分钟，将示例最大值设置为1,000：

```
logman create counter MemCounterCollector -c "\memory\available mbytes"
-si 00:10:00 -sc 1000
```

下面的实例对现存的计数器进行了修改，将示例间隔设置为1小时，将最大示例数设置为10,000：

```
logman update counter MemCounterCollector -si 01:00:00 -sc 10000
```

通过-Cf参数，可以指定计数器文件，其中包含想要从其中收集数据的计数器列表。在计数器文件中，每个计数器应该在单独的一行。示例8-2中，展示了如何通过Logman create counter命令操作计数器文件。在该实例中，计数器文件存在于当前目录，但也可以通过全文件路径来指定处在其他目录下的计数器文件。

示例8-2 收集性能数据

Command line

```
logman create counter GenPerformanceDataCollector -cf collector.txt
-b 06/15/2009 05:30PM -e 06/17/2009 09:30PM
-si 00:10:00 -sc 1000
```

Source for Collector.txt

```
\memory\% Committed Bytes In Use
\memory\Available MBytes
\memory\Cache Bytes
\memory\Cache Bytes Peak
\memory\Committed Bytes
\memory\Commit Limit
\memory\Page Faults/sec
\memory\Pool Nonpaged Bytes
\memory\Pool Paged Bytes
```

8.3.4 配置性能计数器警报

你可以配置警报，以便在特定事件发生或特定的性能指标达到时得到必要的通知。你可以将这些警报作为事件发送到应用程序事件日志，或者将这些警报配置为用于启动任务与性能日志。

要对警报进行配置，应该遵循如下步骤。

(1) 在可靠性和性能监视器中的数据收集器集节点下，鼠标右击左面板的“用户定义”节点，单

击“新建”，之后选择“数据收集器集”。

(2) 在“创建新的数据收集器集”向导中，键入数据收集器名，比如**Processor Usage Alert**。之后选择手工创建（“高级”）选项，单击“下一步”。

(3) 在“创建新的数据收集器集”页面，选择“性能计数器警报”选项，单击“下一步”。

(4) 在“创建新的数据收集器集”页面，单击“添加计数器”对话框，使用“添加计数器”对话框来添加计数器并触发警报。完毕后，单击“确定”。

(5) 在性能计数器面板，选择第一个计数器，之后使用警报条件这一文本框来设置该计数器警报的触发条件。警报触发的时机可能是计数器大于或小于某个特定值，选择大于或小于条件，之后设置触发值。计数单位则取决于对当前选定的计数器是否有效。比如，如果设置为在处理器时间超过98%时产生告警，则应该选中“大于”，键入**98**。对其他计数器，重复这一过程。

(6) 在“创建新数据收集器集”页面上，身份文本框将<默认>作为用户，表示警报将以默认系统账号的权限运行。如果希望以其他用户的权限运行告警，单击“更改”，键入指定账号的用户名与密码，之后单击“确定”。用户名可以以域\用户的格式输入，比如，对Adatum域的WilliamS账号，可以输入**adatum\williams**。

(7) 选择“打开该数据收集器集的属性”选项，之后单击“完成”。这将保存数据收集器集，关闭向导，并打开相关的属性对话框。

(8) 默认情况下，警报是自动启动的。要在选定的性能计数器上为警报设置时间间隔，可以单击“计划”选项卡，单击“添加”。之后就可以设置活动的范围、开始时间以及警报的运行天数。

(9) 默认情况下，只有在计划中设置了截止日期时，才会在条件达到时停止警报。使用“停止条件”选项卡中的选项，可以将事件收集配置为在经过指定的时间间隔（比如7天，或者在事件日志达到设定的最大值）后手工终止。

(10) 完成了对报警计划与停止条件的设置后，单击“确定”。你可以按照8.3.1节中的讲解来对数据收集器进行管理。

(11) 在“可靠性和性能监视器”中，在左面板上选择用于收集告警信息的数据收集器，鼠标右击收集器并选择“属性”。在“警报”选项卡上，可以根据需要配置用于告警的性能计数器。

(12) 在“警告操作”选项卡上，选择“将项记入应用程序事件日志(L)”复选框将警告写入到应用程序事件日志。如果希望在警告触发时启动其他数据收集器，可以使用“启动数据收集器集”列表来选择该数据收集器。

(13) 在“报警任务”选项卡上，可以选择在某报警触发时要运行的Windows Management Instrumentation (WMI) 任务，并指定任务启动时需要传递的参数。

(14) 完成设置后，单击“确定”。

在命令行中，可以使用Logman create alert命令来创建用于性能警告的数据收集器，并使用Logman update alert命令来更新数据收集器的设置。如表8-5中所总结的，这两条命令有类似的参数集，对很多参数，可以使用多余的破折号来否定以前设置的值。比如，如果需要停止向应用程序日志写入告警事件，可以使用--EI。

表8-5 Logman create alert命令与Logman update alert命令的参数

参 数	描 述	可否逆选?
-a	将输出附加到现存的日志文件	是

(续)

参 数	描 述	可否逆选?
-b<mm/dd/yyyy h:mm:ss[AM PM]>	对数据收集器进行调度, 使其在特定的时间开始	
-cnf <[[hh:]mm:]ss>	当指定的时间间隔已到, 或日志文件达到最大值时, 创建一个新文件	是
-config<filename>	创建设置文件, 其中包含了命令选项	
-e<mm/dd/yyyy h:mm:ss[AM PM]>	对数据收集器进行调度, 使其在指定的时间终止	
-el	激活事件日志报告	是
-m <[start][stop][[start][stop] [...]]>	将自动启动或终止改变为手工启动或终止	
-max <value>	以MB为计数单位设置日志文件最大值, 或SQL日志中记录数	是
-o <path dsn!log>	指定输出日志文件的路径, 或SQL数据库中的DSN与日志集名	
-ow	重写现存的日志文件	是
-r	每天会在指定的时间段内重复执行数据收集器	是
-rc <command>	每次日志关闭时运行指定的命令	是
-rdcs <collector>	标识某告警触发时运行的数据收集器	是
-rf <[[hh:]mm:]ss>	为数据收集器设定运行时间	
-s <computer>	在指定的远程计算机上执行命令	
-si <[[hh:]mm:]ss>	设定性能计数器的示例间隔	
-targ	标识在某告警触发时传递给要运行的任务的参数	是
-th <path>thr[<path>thr[...]]	标识要收集的性能计数器与告警阈值, 使用<表示在小于某个值时触发告警, >表示在大于某个值时触发告警	
-tn <task>	标识在警报触发时运行的任务	是
-u <user> [<password>]	为远程计算机设置登录用户名以及口令(可选的)。如果没有设定口令, 在实际应用中会遇到要求输入口令的提示信息	是
-v <nnnnnn mmddhhmm>	为日志名结尾添加版本信息	是
-y	对所有提示信息应答“是”, 从而不会再遇到提示信息	

为性能警报创建数据收集器的基本语法是:

```
logman create alert DataCollectorName -th Counter>Threshold
```

其中, *DataCollectorName*是为数据收集器设置的唯一性的名称, *Counter*用于设置相对计数器路径, *Threshold*则设置了告警阈值。进行阈值设置时, 使用<表示在低于某个阈值时产生告警, >则表示在大于某个阈值时产生告警。比如, 如果需要在处理器使用率超过98%时产生告警, 就可以使用如下命令:

```
logman create alert ProcessorAlert
-th "\Processor(_Total)\% Processor Time>98"
```

如果需要在可用内存小于64M时产生告警, 则可以使用如下命令:

```
logman create alert LowMemAlert -th "\Memory\Available Mbytes<64"
```

通过-Th参数, 可以指定多条告警。下面的实例中, 只需要使用空格将告警分隔开:

```
logman create alert CoreAlerts
-th "\Processor(_Total)\% Processor Time>98" "\Memory\Available Mbytes<64"
```

要注意的是, 以上面的方式创建告警并不能启动性能警报。要启动性能警报, 或者像以前讨论的那样手工启动与终止, 或者对其进行调度, 使其在某个时间段内运行(此时间段必须晚于当前日期和

时间)。下面的实例中，在5/10/2009的上午7:30启动数据收集器：

```
logman create alert ProcessorAlert
-th "\Processor(_Total)\% Processor Time>98"
-b 05/10/2009 07:30AM
```

如果设置了错误的开始时间、结束时间，或者二者同时设置错误，可以使用Logman alert update命令对其进行修改。如果只设置开始时间，而没有设置终止时间，则数据收集器会一直运行下去。

如果需要将告警事件写入到应用程序日志，可以使用-EI参数激活事件日志报告功能。如果创建数据收集器时没有指定该参数，则可以对数据收集器进行更新。下面的命令会激活事件日志报告功能：

```
logman update alert ProcessorAlert -ei
```

下面的命令会禁用事件日志报告功能：

```
logman update alert ProcessorAlert --ei
```

8.3.5 查看数据收集器报告

进行故障排除时，你可能需要记录一段时间间隔之内的性能数据，并通过对数据的分析寻找故障的可能原因。对每一个曾经使用过或当前处于活跃状态的数据收集器，都可以发现相关的数据收集器报告。对于数据收集器集本身，数据收集器报告通常被组织为两个通常的范畴：用户自定义与系统定义。

如果需要在可靠性和性能监视器中查看数据收集器报告，可以扩展报告节点，之后扩展需要分析的数据收集器的单独的报告节点。在数据收集器的报告节点下，可以发现每一个日志会话的单独的报告，每一个日志会话在日志启动时开始，在日志终止时结束。

日志的编号是顺序递增的，最近期的日志具有最大的日志编号。要以图形化的格式来查看日志并对其相关数据进行分析，可以双击该日志。要记住的是，如果数据收集器正在进行事件记录，则无法查看最近期的日志。要停止数据收集，可以鼠标右击该数据收集器，之后选择“终止”。默认情况下，收集的数据是以图形化的视图展示的，包含了从收集开始到收集结束这一过程中收集到的所有数据。此外，只有选定的用于记录的计数器才是可用的。如果某报告不包含要使用的计数器，就需要修改数据收集器属性，重启日志过程，之后重新检查日志。

通过如下步骤，可以修改报告的详细资料。

(1) 在可靠性和性能监视器中，右击性能监视器节点，选择“属性”，在“性能监视器属性”对话框中，单击“来源”选项卡。

(2) 指定要分析的数据源。在数据源复选框下，选择日志文件，单击“添加”打开“选择日志文件”对话框，之后可以选择要分析的日志文件。

(3) 指定要分析日志的时间范围。单击“时间范围”，之后拖动“全部范围”进度条来指定适当的开始时间与结束时间

(4) 单击“数据”选项卡，之后可以选择要查看的计数器。要从图形化视图中删除计数器，可以选定该计数器，之后单击“删除”。单击“添加”可以显示“添加计数器”对话框，之后可以选择要分析的计数器。

(5) 单击“确定”，在监视器面板中，单击“更改图表类型”按钮来选择图形类型。

在命令行中，通过使用Tracerpt命令，也可以生成类似的报告。Tracerpt命令可以对数据收集器日志进行处理，并根据其中的事件生成踪迹分析报告与转储文件。表8-6中总结了Tracerpt命令的可用参数。

表8-6 Tracerpt命令的参数

参 数	描 述
-config <filename>	指定包含命令选项的设置文件
-df <filename>	为微软特定的计数/报告框架文件（应该用于处理踪迹）设定名称
-export <filename>	为事件框架导出文件设定名称，默认为Report.xml
-f <XML HTML>	将报告文件格式设置为.xml或.html
-gmt	将时间戳转化为GMT时间
-i <path>	设置提供者镜像路径，每个路径值以分号分隔开
-int <filename>	为用于解释型事件结构的转储文件设置名称
-lr	设置限制性较弱的输出信息，以便没有与框架匹配的事件可以用尽量服务的方法进行显示
-o <filename>	设置文本输出文件，用于写入分析后的数据。默认为Default.xml
-of <CSV EVTX XML>	将转储文件格式设置为.csv、.evtx、.xml
-pdb <path>	设置符号服务器路径，每个路径值以分号分隔开
-report <filename>	设置文本文件名，用于写入数据的详细分析报告。默认为Workload.xml
-rl <level>	设置系统报告级别，取值为1~5。默认为1
-rt <session_name[session_name[...]]>	设置实时事件踪迹会话数据源，而不使用转换的日志文件
-rts	从事件踪迹头中提取原始时间戳并将其添加到输出，只能与-O一起使用，不能与-Report或-Summary一起使用
-summary <filename>	设置文本文件名，用于写入数据分析的总结报告。默认为Summary.txt
-tmf <filename>	设置Trace Message Format (TMF) 定义文件名，用于处理踪迹
-tp <path>	设置TMF文件搜索路径，每个路径值以分号分隔开
-y	对所有提示问题回答“是”，从而不会再看到提示信息

使用Tracerpt的一种方法是指定要使用的数据收集器日志名。默认情况下，数据收集器日志被写入到受约束的子目录C:\PerfLogs。因此，如果该目录中某日志名称为DataCollector01.blg，就可以使用如下命令对其进行分析：

```
tracert "C:\Perflogs\Admin\Process Monitor\000001\DataCollector01.blg"
```

上面命令执行后，会在当前目录下创建2个文件：Dumpfile.xml，用于写入分析后的输出信息；Summary.txt，用于写入总结报告。要获取详尽的报告与相关联的框架文件，可以使用-Report参数与-Export参数，如下面命令所示：

```
tracert "C:\Perflogs\Admin\Process Monitor\000001\DataCollector01.blg"
-report -export
```

上面命令执行后，会在当前目录下创建4个文件：Dumpfile.xml，用于写入分析后的输出信息；Summary.txt，用于写入总结报告；Workload.xml，用于下如详尽的报告；Report.xml，用于写入事件框架报告文件。

你也可以指定用于输出的文件，如下面实例所示：

```
tracert "C:\Perflogs\Admin\Process Monitor\000001\DataCollector01.blg"
-o c:\PerfLogs\dumpfile.csv -summary c:\PerfLogs\summary.txt
-report report.txt
```

作为管理员，每天都会重复执行一些相同或相似的任务。其中某些任务有时候还不得不提前或推迟到非工作时间进行。其中一些任务可能是例行的维护活动，比如删除临时文件（以免磁盘空间耗尽），或者备份重要数据等。另外一些任务可能是比较繁琐的过程，比如，在所有交易服务器上搜索事件日志，以便发现需要解决的问题。幸运的是，如果将这些任务分解为一系列步骤，就可以自动执行这些任务，Windows提供了下面两种实现方法。

- **Schtasks**。一个高级命令行工具，可根据计划来运行命令、脚本以及程序。你可以根据实际需要任务进行计划，使其只运行一次，每分钟运行一次，在特定的时间间隔后运行（比如每小时、每天、每星期、每月运行一次），在系统启动时运行，在登录时运行或在系统空闲时运行。
- **任务计划程序**。一个图形界面工具，可根据调度计划来运行命令、脚本以及程序。任务计划程序与Schtasks执行同样的操作，二者可以一起使用，也可以分别管理或执行另外一款工具创建的任务。

Schtasks命令与任务计划程序可以互换使用，本章讨论如何使用这两款工具来自动运行程序、命令行工具与脚本。大多数情况下，你会发现，同时理解这两款工具是有用的，即便有时候可以通过任务计划程序单击与选择，但你也可以使用命令行来完成同样的任务。

9.1 在本地与远程系统上执行计划任务

在命令行中运行的一切对象都可以配置为计划任务，包括命令行工具、脚本、应用程序、快捷方式、文档等，也可以在其中指定命令行参数。有些时候，你可能需要在当前登录的本地系统上进行这些任务计划与配置，但更常见的情况是从本地系统对网络中的远程系统进行这些操作。

9.1.1 计划任务简介

本地系统与远程系统上的计划任务都是通过Task Scheduler服务激活的，对于需要在其上执行计划任务的任意系统，都必须首先启动该服务。默认情况下，Task Scheduler服务以本地系统账号登录系统，该账号通常不具备执行管理型任务的许可权限。为此，对要计划执行的每一个任务，都应该对其进行适当配置，使用具有适当权限的账号运行相应任务。此外，对需要在其上执行计划任务的所有系统，要确保Task Scheduler服务配置为自动启动。再次强调，必须对Task Scheduler服务的启动方式与登录账号选项进行正确配置。

你可以使用任务计划程序控制台来查看并操作计划任务。要访问任务计划程序，可以依次单击“开始”、“管理工具”、“任务计划程序”。任意用户都可以在本地计算机上设置计划任务，并根据需要对其进行查看与修改。管理员可以在本地计算机上计划、查看、修改所有任务（那些受限制的系统任务除外）。如果需要在远程计算机上计划、查看、修改任务，则必须具备该远程计算机上管理员组成员的身份，或者在遇到提示信息时提供管理员登录凭据。

在任务计划程序中，任务计划程序库包含了计算机上定义的所有任务。与以前的Windows版本不同的是，Windows Vista与Windows Server 2008都广泛地使用了计划任务，很多任务在第一次安装操作系统时就已经自动创建。安装角色、角色服务、功能以及应用程序时，这些组件也可能创建额外的任务。在图9-1中，任务计划程序库展示了如下一些要素。

- **名称**。表示任务名，在形式上可以是任意字符串。与任务的其他属性一样，任务名可以在创建时设置。
- **状态**。表示任务当前状态。“正在运行”表示任务计划程序已启动该任务，该任务正在运行中。“准备就绪”表示该任务已激活，准备就绪，并等待触发。“禁用”表示该任务被禁用，不再运行。“失败”表示该任务由于某些故障无法启动。
- **触发器**。列出了与该任务相关联的触发器。
- **下次运行时间**。表示该任务下次将要运行的日期与时间。Never表示该任务在调度运行的时间过后不会再次运行，这种情况通常是一次性任务。
- **上次运行时间**。表示该任务上次运行的日期与时间，Never表示该任务尚未运行过。
- **上次运行结果**。表示退出错误代码。为0时表示没有发生错误，其他数值表示发生了某种类型的错误。
- **创建者**。表示创建该任务的用户名。
- **创建时间**。表示该任务创建的日期与时间。

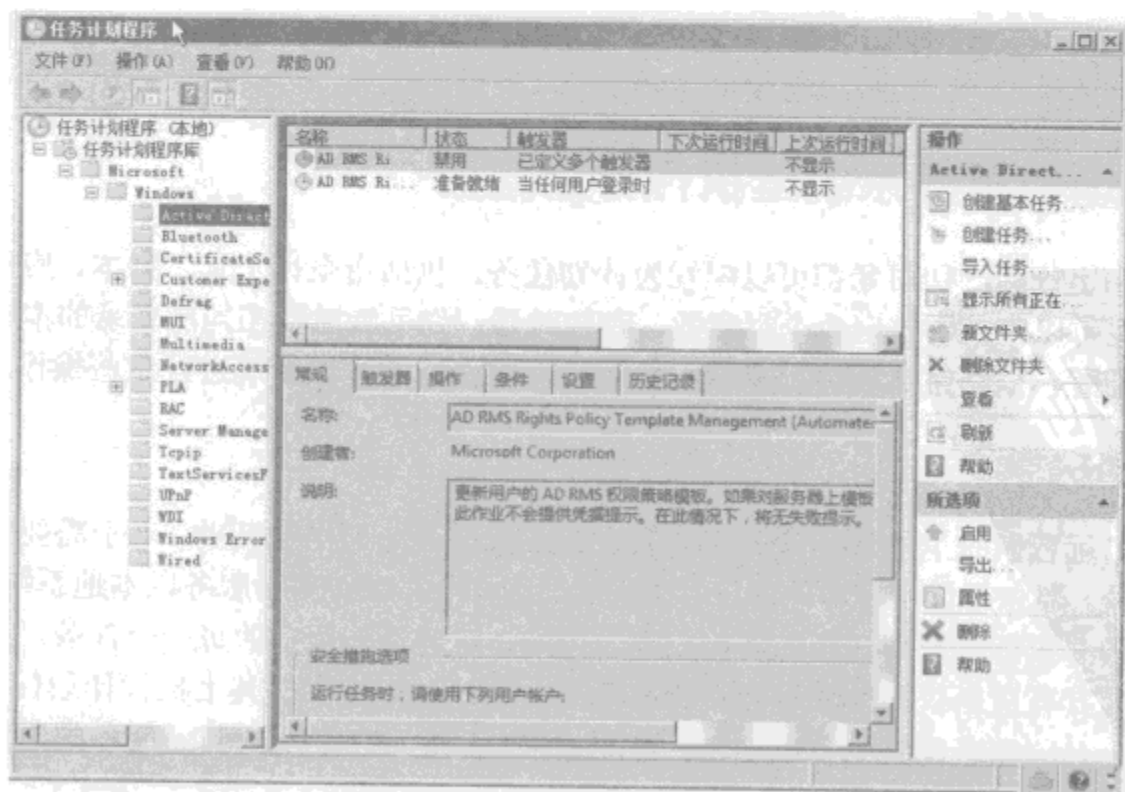


图9-1 使用任务计划程序在图形界面中操作计划任务

Windows Vista与Windows Server 2008中包含两种通常意义的任务类型。

- **标准任务。**自动执行的例行任务，执行日常的维护操作。这些任务对用户可见，并可以在必要时进行修改。
- **隐藏的任务。**自动执行的特定系统任务。默认情况下，这些任务对用户不可见，大多数时候也不应该进行修改。有些隐含任务是通过相关程序进行创建与管理的，比如Windows Defender。

注解 在任务计划程序的“查看”菜单中，选择“显示隐藏的任务”，可以显示隐藏的任务。

在任务计划程序库中，Microsoft\Windows下与Microsoft\Windows Defender下的任务为系统任务。Microsoft\Windows下的任务主要用于完成计算机上很多后台的维护操作，Microsoft\Windows Defender下的任务则用于自动扫描恶意软件。Windows Defender是Windows Vista中默认使用的恶意软件扫描工具，在Windows Server 2008中，通过安装桌面体验功能，也可以将该工具添加到系统中。

尽管可以在任务计划程序库中的任一层次创建任务，但大多数情况下是在其顶层进行任务创建。为做到这一点，创建任务时，可以选择任务计划程序节点而非其他低层节点。

如果需要对任务进行组织，可以在任务计划程序库中创建文件夹。这些文件夹在形式上成为任务计划程序库层级结构中的节点，并充当任务容器的角色。在大型企业网络中，甚至可以在任务计划程序库中创建子层级结构。通过如下步骤，可以创建文件夹，使其包含自己的任务。

(1) 在任务计划程序中，选择用于包含文件夹的顶层节点。比如，如果希望文件夹成为任务计划程序库节点的子节点，则可以选择该节点。

(2) 在“操作”菜单或“操作”面板中，选择“新文件夹”。在弹出的对话框中，为该文件夹输入一个唯一的名字，之后单击“确定”。

在Windows Vista与Windows Server 2008中，任务计划程序是与操作系统中实现的Windows安全模型完全整合在一起的。定义任务时，需要指定运行该任务的用户账号。默认情况下，任务以用户账号的标准权限运行。如果任务需要更高的权限，而又不希望Windows阻止其运行并显示用户账号控制提示，就需要在定义任务时指定Windows总是使用最高权限运行该任务。以用户账号的最高权限运行时，任务计划程序会弹出提示要求输入该用户账号的口令，之后，该口令会被安全地存储在系统中。

注解 在修订后的任务计划程序中，微软改变了任务使用用户登录凭据的方式。以前，如果某用户账号的口令进行了修改，就需要对使用该用户账号的任务的口令设置进行修改。在Windows Vista与Windows Server 2008中，如果任务只访问本地资源，则不需要在用户口令改变后进行任务相关的口令更新。口令更改不会影响任务，任务可以继续运行。如果任务访问外部资源，则需要用户在用户账号口令改变之后，对使用该账号的任务相关口令设置进行更新。

任务包含了很多属性，主要包括下面5个。

- **触发器。**指定了任务开始与结束的环境。
- **操作。**定义了任务启动时执行的操作。
- **条件。**限定了任务启动与终止的条件。
- **设置。**影响任务的行为。
- **记录。**展示了任务运行或尝试运行时生成的事件。

与以前的Windows版本中的计划任务不同的是，在Windows Vista与Windows Server 2008中可以为

任务指定不止一个触发器与动作。这意味着，每个计划任务可以运行多个程序、工具或脚本，也可以被配置为以多种方式运行，包括下面4种。

- 在特定的时间与日期运行，比如，在2009年10月25日的下午5:45运行。
- 在指定的时间间隔运行，比如，在每个星期一、星期三、星期五的下午5:45运行。
- 在特定系统事件发生时运行，比如，在某用户登录系统时运行。

以上面几个运行条件的任意合理组合运行，或以其他未列出的方式运行。

在任务的相关属性中，任务触发器应该引起特别的关注，因为他们并不总是能按预期的那样工作，任务触发器包括下面5个。

- **系统启动**。如果将计划任务设置为在系统启动时运行，则任务计划程序会在系统每次启动时自动运行该任务（而不需要用户的交互），并一直运行到任务完成、任务被终止，或者系统关机。要记住的是，只有任务的属主或管理员才可以终止运行中的任务。
- **系统登录**。如果需要将计划任务设置为在某用户登录系统时运行，则可以对其进行配置，使其在任意用户登录系统时运行（而不管是哪个用户对其进行配置），或者只有在某个指定的用户登录系统时运行。经过这样配置并启动后，任务计划程序会一直运行该任务，直至该任务完成、任务被终止，或者该用户退出登录。这种方式运行的任务可以是交互式的，也可以是非交互式的，取决于其具体配置方式。

提示 如果某用户使用自己的登录凭据配置了一个交互式的任务，但其他用户登录了系统，则该任务仍然会以其创建者的许可权限运行，并且在其他用户退出登录时并不会终止运行（因为该任务由其创建者拥有，其他用户不具备权限终止其运行）。进一步地，使用快速用户切换时，登录型任务并不会在用户切换后运行，只有在某用户登录而其他用户都退出登录时才运行。

- **系统空闲**。如果将计划任务设置为在系统空闲时运行，则任务计划程序会在某个时间间隔之内没有任何用户活动后执行该任务。比如，你可以创建一个任务，使其在系统空闲5分钟之后运行。要记住的是，后续的用户活动并不会终止该任务。该任务会一直运行，直至任务完成或被终止。
- **Windows事件**。如果将计划任务设置为在特定Windows事件发生时运行，则任务计划程序会在Windows将带有指定标识符的事件写入到特定事件日志时运行该任务。在基本的配置中，可以指定一个单独的事件标识符、一个可选的事件源以及要进行监控的单一事件日志。在自定义配置中，可以定义一个事件过滤器，并像前面章节中讲述的那样对其进行操作，可以实现对多个日志、多个事件源以及两者相结合的多种事件类型的监控。
- **用户会话**。如果将计划任务设置为在某用户建立了一个终端服务用户会话时运行，则任务计划程序会在用户从本地或远程系统建立连接、创建用户会话时运行该任务。你也可以将计划任务设置为在用户终止来自本地或远程系统的终端服务会话时运行。

尽管可以为任务定义多个操作，但实际上也可以在任务启动时创建一个命令行脚本，使其运行多个命令、程序与工具，以便执行其他任务。你可能需要该脚本以特定用户或管理员的登录凭据运行，以便保证该脚本有必要的许可权限与访问权限。该脚本还应该配置其他必需的用户设置，以便确保所做的一切操作都在其可控范围之内。此外，域用户设置，比如驱动器映射，在必要的时候也是可用的。

真实场景 配置任务的运行方式时,可以指定该任务运行时使用的用户账号与登录口令。对于递归的任务,这种做法会导致问题,尤其是在许可权限与口令变更时更是如此,而这种变更通常是难以避免的。如果账号许可权限或口令发生变更,而某些任务又需要对远程资源进行操作,则至少需要对使用这些登录凭据的一个任务的属性进行编辑,并为该账号提供新的登录凭据。

9.1.2 监控计划任务

任务计划程序不会对用户提供的信息进行验证,也不会对程序、命令、工具的可用性进行验证。如果提供了错误的信息,则任务只是不能运行或者运行时出错。如果需要对任务进行检查,有一种方法是在任务计划程序中查看任务状态以及上次的运行结果,这些信息是任务上次运行时生成的。如果上次运行结果为错误信息,就需要解决存在的问题,以便任务可以正常运行。通过单击任务在任务计划程序中的入口,就可以对其属性进行查看与编辑。

列为正在运行状态的任务实际上也可能并未运行,而是一个失去响应的或失控的进程。你可以使用上次运行时间来检查失去响应的或失控的进程,该值用于表明任务的启动时间。如果检查结果表明该任务已运行超过一天,则很可能存在需要纠正的问题。如果是脚本,则可能在等待输入信息,或者读写文件存在问题,也可能是一个需要终止的进程。要终止任务,可以在任务计划程序中右击该任务,之后选择“结束”。

然而,上次运行结果并不能表明上一次运行该任务之前是否发生过问题。要深入挖掘并透彻理解任务的运行历史,你应该定期检查任务计划程序的运行日志。在事件查看器中,该日志存储于 Applications And Services Logs\Microsoft\Windows\TaskScheduler\Operational。对任务计划程序日志进行检查时,会发现如下一些信息。

- 记录了Task Scheduler服务启动与退出(终止)的时间。
- 记录了任务启动、完成运行的时间,以及退出错误代码或结果代码。取值为0时意味着该任务正常运行和终止,其他值都表示发生了某种错误。
- 记录了任务计划程序启动任务时产生的告警与错误消息。

提示 在命令行中,任务计划程序操作日志的引用名为\MicrosoftWindows-TaskScheduler\Operational。默认情况下,该日志存储在%SystemRoot%\System32\Winevt\Logs文件夹下,名为Microsoft-Windows-TaskScheduler\Operational.evtx。通过使用第6章、第8章中讨论的技术,可以像对其他日志一样操作这一日志。

任务计划程序中包含了操作日志的自定义过滤视图。选定了任务计划程序中的任务计划程序节点后,主面板分为几个子面板,包括任务状态子面板与活动任务子面板。通过任务状态面板中的选项,可以查看不同时间间隔内(最近1个小时、最近24个小时、最近7天、最近30天)所有计划任务的状态摘要。通过活动任务面板,可以查看运行中任务的摘要信息。通过双击一个运行中的任务,就可以访问任务计划程序库中的任务定义。

选定了任务计划程序中的任务定义之后,单击“记录”选项卡,就可以查看任务计划程序操作日志的筛选后的视图,其中包含了与该任务相关的事件。如果选中其中一个特定的事件,就可以在下一级面板中看到该事件的详细信息。

脚本运行时,如果需要对所发生的事件有更详尽的理解,可以将命令与工具的输出信息记录到单独的日志文件中,之后可以通过该日志文件分析那些命令与工具是否生成了预期中的结果。通过对标准输出与标准错误输出的重定向,可以将命令输出信息写入到指定的文件。下面的实例中,DEFRAG命令的输出信息以及错误信息被以添加模式写入到Stat-log.txt。

```
defrag c: >> c:\logs\stat-log.txt 2>&1  
defrag d: >> c:\logs\stat-log.txt 2>&1
```

警告 像这些实例中那样操作目录时,目录本身必须是存在的。如果不存在,系统并不会自动创建该目录,由于目录不存在导致的错误也不会写入到日志文件。

真实场景 将命令输出信息写入到日志文件并不能帮你解决可能发生的每一个问题,但对于了解任务是否按预期进行还是很有用的。如果需要对问题进行故障排除,要记住的是,导致任务失败的原因有很多,其中有些原因超出了可控范围。比如,如果在计划任务的执行时刻系统关机了,则任务自然无法运行。如果需要保证任务在错过了调度的开始时间后仍能运行,可以在任务的属性对话框的“设置”选项卡选择“如果过了计划开始操作,立即启动任务”。激活这一功能后,如果因为系统关机导致错过了计划任务的开始时间,则Windows会尽可能快地运行该任务。

9.2 使用任务计划程序计划任务

你可以使用任务计划程序创建基本任务与高级任务。基本任务包括触发器与操作,用于快速设置常见的计划任务。高级任务包括触发器、操作、条件、设置,主要提供给高级用户或管理员使用。

9.2.1 创建基本任务

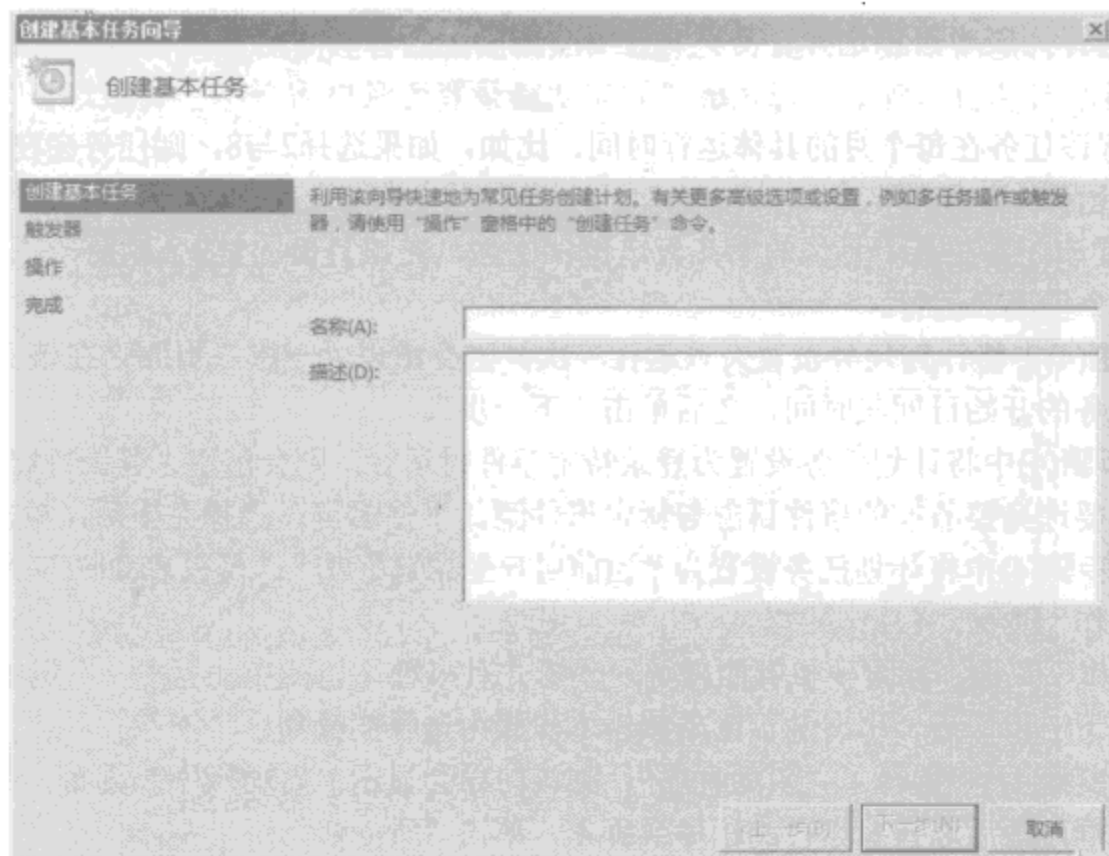
基本任务包含很多默认的设置,易于创建。默认情况下,某用户创建的基本任务以该用户的登录账号运行,并且只有在该用户登录的情况下才运行。这些任务以标准用户权限运行,而不会以最高权限运行。出于对兼容性的考虑,这些任务在配置上同时兼容Windows Vista与Windows Server 2008。此外,这些任务只有在计算机处于AC电源模式下才会启动,当计算机切换到电池组模式时会终止运行。另外一种情况是,基本任务运行超过3天后会终止运行。

通过完成如下步骤,可以创建一个基本任务。

(1) 依次单击“开始”、“管理工具”、“任务计划程序”,启动任务计划程序。如果需要查看远程计算机上的日志,在控制台树(左面板)上鼠标右击任务计划程序入口,选择“连接到另一台计算机”。之后,在“选择计算机”对话框中,输入想要访问的计算机的主机名或IP地址,单击“确定”。

(2) 在任务计划程序库中的任一层次,都可以创建基本任务。其方法是:右击需要存储待创建任务的节点,选择“创建基本任务”,之后将启动“创建基本任务”向导。

(3) 在图9-2中所示的“创建基本任务向导”页面,为待创建任务键入任务名及其描述信息。任务名应该简短而富于描述性,以便根据名称确定其功能。可选的描述信息中可以对该任务的功能目标进行详尽的解释。完毕后,单击“下一步”。



(4) 在“任务触发器”页面，为该任务选择运行方式。你可以将该计划任务设置为只运行一次，周期性运行（每天、每星期，或每个月运行一次），计算机启动时发生特定事件后运行，该任务的创建者登录时运行。完毕后，单击“下一步”，之后的向导页面依赖于所选择的任务调度时间与方式。

(5) 如果步骤(4)中将计划任务设置为每天运行一次，则会出现“每日”页面。在该页面中，可以通过如下字段对该任务进行配置。

- ☐ **开始。**该选项用于设置任务的启动日期与时间。
- ☐ **通用时间。**该选项根据格林尼治平均时间（GMT）来对任务进行调度，而非根据本地时间。GMT是英国伦敦格林尼治时区，子午线时间。
- ☐ **每隔。**该选项用于将计划任务设置为每天运行、每隔一天运行或每隔 n 天运行，开始时间则根据具体设置。比如，如果希望该任务每隔一天运行一次，就可以将“每隔……天发生一次”文本框设置为2天。

上述配置完成后，单击“下一步”。

(6) 如果步骤(4)中将计划任务设置为每周运行一次，则会出现“每周”页面。在该页面中，可以通过如下字段对该任务进行配置。

- ☐ **开始。**该选项用于设置任务的启动日期与时间。
- ☐ **通用时间。**该选项根据格林尼治时间（GMT）来对任务进行计划，而非根据本地时间。
- ☐ **每隔。**该选项用于将计划任务设置为每星期运行、每隔一星期运行或每隔 n 星期运行。
- ☐ **每周的哪几天。**设置该任务在每星期的具体运行时间，比如星期二，或星期二与星期五。

(7) 如果步骤(4)中将计划任务设置为每个月运行一次，则会出现“每月”页面。在该页面中，可以通过如下字段对该任务进行配置。

- ☐ **开始。**该选项用于设置任务的启动日期与时间。

- 月。该选择列表中可以选择任务具体在哪个月份运行，可以选择所有月份，也可以一个或几个月份。
- 天。设置该任务在每个月的具体运行时间。比如，如果选择2与8，则任务会在该月份的第2天与第8天运行。
- 在。设置该任务在某个月份的第几天出现，比如，在该月份的第2个星期一，或该月份的第1个与第3个星期二运行。

(8) 如果步骤(4)中将计划任务设置为只运行一次，则会出现“一次”页面。在该页面中，使用开始选项设置该任务的开始日期与时间，之后单击“下一步”。

(9) 如果在步骤(4)中将计划任务设置为登录特定事件时运行，则会出现“登录特定事件时”页面。在该页面中，需要选择要监控的事件日志与特定事件源、事件ID等，选择完毕后，单击“下一步”。

(10) 如果在步骤(4)中将计划任务设置为“当前用户登录时”或“计算机启动时”，单击“下一步”，之后会出现“操作”页面。

(11) 在“操作”页面，指定要执行的任务。其形式有多种，包括启动程序、发送电子邮件、显示消息等。单击“下一步”后，出现的页面依赖于本步骤中选择的操作。

(12) 如果在步骤(11)中选择了“启动程序”，则该向导会显示“启动程序”页面。单击“浏览”显示“打开”对话框，之后选择要运行的程序或脚本，单击“下一步”。

(13) 如果在步骤(11)中选择了“发送电子邮件”，则该向导会显示“发送电子邮件”页面。在电子邮件表单中，填充电子邮件消息的发件人、收件人、主题、正文等字段，就可以实现对电子邮件自动发送的配置。在SMTP服务器文本对话框中，输入用于传输电子邮件的邮件服务器的完全限定域名，单击“下一步”。

(14) 如果在步骤(11)中选择了“显示消息”，则该向导会显示“显示消息”页面。在其中可以对消息进行配置，使其在任务启动时显示于桌面。在文本框中，输入消息标题与内容，单击“下一步”。

(15) 在摘要页面，查看任务相关的详细资料，确认后单击“完成”按钮。任务创建后，通过其属性对话框，可以修改基本任务的默认设置。

9.2.2 创建高级任务

对高级任务，可以直接在与任务的标准属性对话框类似的对话框中进行设置。高级任务有一些与基本任务类似的默认的初始设置。

通过完成如下步骤，可以创建一个高级任务。

(1) 依次单击“开始”、“管理工具”、“任务计划程序”，启动任务计划程序。如果需要查看远程计算机上的日志，在控制台树（左面板）上鼠标右击任务计划程序入口，选择“链接到另一台计算机”。之后，在“选择计算机”对话框中，输入想要访问的计算机的主机名或IP地址，单击“确定”。

(2) 在任务计划程序库中的任一层次，都可以创建任务。其方法是：右击需要存储待创建任务的节点，选择“创建任务”，之后将启动“创建任务”向导。

(3) 在图9-3中所示的“常规”选项卡中，为待创建任务键入任务名及其描述信息。任务名应该简短而富于描述性，以便根据名称确定其功能。可选的描述信息中可以对该任务的功能目标进行详尽的解释。完毕后，单击“下一步”。

(4) 默认情况下，高级任务也以其创建者的用户账号运行。如果需要以其他用户账号运行该任务，单击“更改用户或组”，之后在弹出的对话框中选择用户或组，使得任务以该用户或组运行。

(5) 默认情况下, 任务只有在创建者或指定的用户登录时才运行。如果需要将其配置为不考虑具体的登录用户, 可以选择“不管用户是否登录都要运行”通过这一设置, 任务计划程序会存储创建者或指定用户的登录凭据。

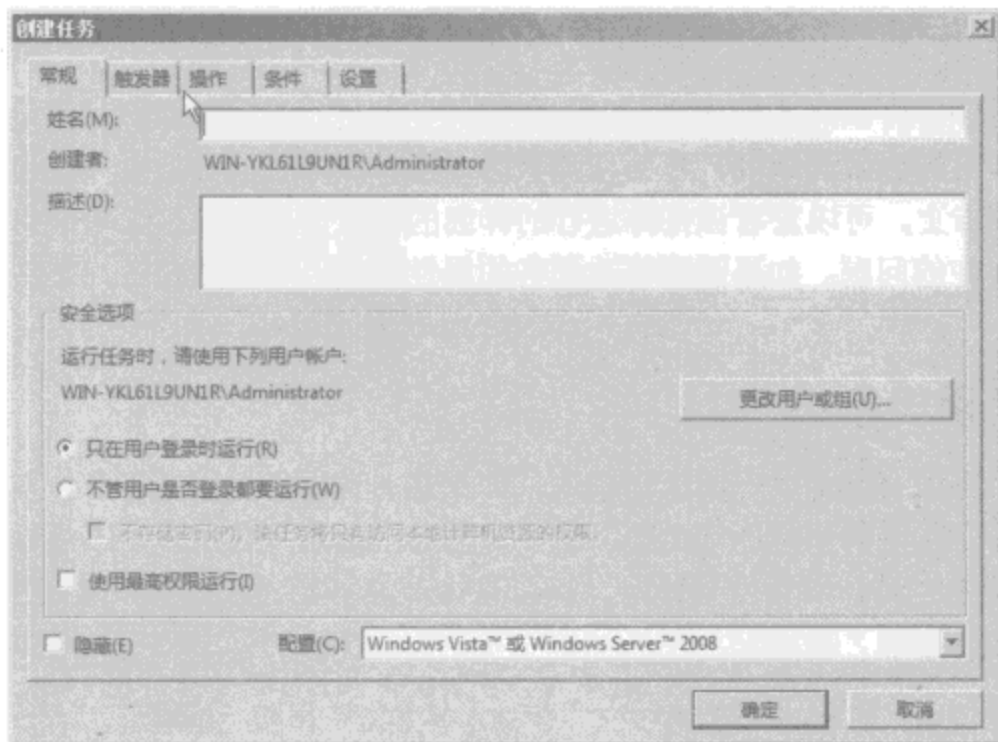


图9-3 使用“创建任务”对话框创建高级任务

注解 如果不希望任务计划程序存储与这些登录凭据相关联的口令, 可以选择“不存储口令”这一复选框。然而, 要记住的是, 这样进行设置时, 虽然不影响任务访问本地计算机的资源, 但会阻止任务访问远程计算机的资源。

(6) 默认情况下, 任务以创建者账号或指定用户账号的标准权限运行, 而非最高可能权限。如果任务需要管理员权限才能运行, 则需要选中“使用最高权限运行”复选框。

(7) 默认情况下, 任务总会在任务计划程序中显示。如果需要将任务隐藏, 可以选中“隐藏”复选框, 这将使得默认的任务计划程序视图中不会显示该任务。如果用户需要查看隐含的任务, 就需要选择显示隐含任务选项。

(8) 默认情况下, 高级任务是为Windows Vista与Windows Server 2008创建与使用的。如果需要创建高级任务并将其导出, 之后再导入到此前版本的Windows系统中, 可以在配置列表中选择Windows Server 2003、Windows XP或Windows 2000。

(9) 在“触发器”选项卡上, 可以使用其上提供的选项来创建与管理触发器。通过触发器, 可以将计划任务设置为只运行一次, 周期性运行(每天、每星期, 或每个月运行一次), 计算机启动时发生特定事件后运行, 该任务的创建者登录时运行。要创建触发器, 单击“新建”, 之后使用可用的选项来对触发器进行配置, 完毕后单击“确定”。根据实际需要, 可以定义多个互不冲突的触发器。

(10) 在“操作”选项卡上, 可以使用其上提供的选项来创建与管理操作。这里, 操作可以是启动一个程序、发送一封电子邮件或显示一条消息。要创建一个操作, 单击“新建”, 之后使用可用的选项来对触发器进行配置, 完毕后单击“确定”。根据实际需要, 可以定义多个互不冲突的操作。

- (11) 在“条件”选项卡上,可以指定启动或终止任务的限定条件。
- (12) 在“设置”选项卡上,可以为任务选择任意附加的可选设置。
- (13) 单击“确定”,完成任务的创建。创建任务后,可以通过其属性对话框对其设置进行修改。

9.2.3 管理任务属性

通过任务计划程序,可以对本地计算机上配置好的任务进行访问与操作。选定一个任务后,通过主面板底部的选项卡可以查看其属性,包括任务状态、上一次运行时间、上一次运行结果等信息。在任务的“记录”选项卡上,可以查看该任务运行的详细的记录信息。通过记录信息,可以了解并有助于解决任务运行中存在的问题。如果需要修改任务的属性,可以双击该任务,之后通过属性对话框进行必要的修改。

9.2.4 激活与禁用任务

根据实际需要,可以对任务进行激活与禁用。如果临时性地不使用某任务,可以对其进行禁用。如果需要使用该任务,就可以对其进行激活。通过对任务的激活与禁用,而不是删除,可以节省下重新配置任务设置所需的时间。要禁用某任务,可以右击该任务,之后选择“禁用”;要激活某任务,可以右击该任务,之后选择“激活”。

9.2.5 将任务复制到其他计算机

任务计划程序中的导入、导出选项使得任务在不同计算机之间的复制变得非常容易。通过如下步骤,可以将任务复制到其他计算机。

- (1) 依次单击“开始”、“管理工具”、“任务计划程序”,启动任务计划程序。
- (2) 如果需要复制的任务不在当前登录的计算机上,右击任务计划程序节点,选择“连接到其他计算机”,使用该对话框连接到待复制任务所在的源计算机。
- (3) 右击待复制的任务,选择“导出”。在“另存为”对话框中,为该任务的XML配置文件选择一个存储位置与文件名,之后单击“保存”。
- (4) 右击任务计划程序节点,选择“连接到其他计算机”,使用该对话框连接到任务将要复制到的目标计算机。
- (5) 右击任务计划程序节点,选择“导入”。在“打开”对话框中,浏览到该任务的XML配置文件所在存储位置,选中并打开该文件。
- (6) 任务计划程序读取配置文件中该任务的原始设置,并打开“创建任务”对话框。通过该对话框,可以修改原始设置,使其满足目标计算机上的需要。完毕后,单击“确定”。

9.2.6 立即运行任务

并不是必须等待任务计划运行的时间到达才能运行任务,实际上,你可以在任意时间运行任务。其方法是:打开任务计划程序,右击要运行的任务,之后选择“运行”。

9.2.7 移除不需要的任务

如果不再需要某个任务,可以将其永久性地删除。其方法是:打开任务计划程序文件夹,右击要删除的任务,选择“删除”。

9.3 使用 Schtasks 设置任务计划

通过Schtasks，可以像使用任务计划程序一样设置计划任务。实际上，命令行形式的Schtasks与图形用户界面（GUI）形式的任务计划程序是等价的，二者可以互相处理对方的计划任务。

Schtasks是命令行中可用的复杂命令之一，包含几种不同的子命令集，下面几节将对如下的一些子命令分别进行讨论。

- Schtasks/Create。用于创建计划任务。
- Schtasks/Change。用于改变现有任务的属性。
- Schtasks/Query。用于在本地计算机或其他指定的计算机上显示计划任务。
- Schtasks/Run。用于立即启动计划任务。
- Schtasks/End。用于终止运行中的任务。
- Schtasks /Delete。用于删除不再需要的任务。

9.3.1 使用 Schtasks/Create 创建计划任务

通过Schtasks /Create，可以创建只运行一次的任务、可重现的任务，以及基于特定系统事件的任务，比如登录事件与启动事件等。使用该命令创建任务的基本语法格式如下：

```
schtasks /create /tn TaskName /tr TaskToRun /sc ScheduleType
[/mo Modifier]
```

其中，*TaskName*为字符串形式的任务名，*TaskToRun*指定了要运行的程序、命令行工具、脚本所在位置的文件路径，*ScheduleType*指定了运行计划，*Modifier*是一个可选的值，用于对运行计划进行修改（根据计划类型）。通过上面的命令语法格式，可以在本地计算机上创建任务，并为其赋予创建者的许可权限。如果未提供账号口令，则创建任务时会弹出提示信息要求输出口令。

表9-1中列出了*ScheduleType*的有效值，要注意不同计划类型可以接受的modifiers以及使用方法，在本章后面部分，将详细讨论每一种调度类型以及modifier。此外，要注意以下几点。

- 你可以以逗号分隔的列表形式输入某星期的具体几天，比如Mon,Wed,Fri。也可以使用连字符（-）指定连续的几天，比如Mon-Fri，代表周一到周五。
- 你可以以逗号分隔的列表形式输入某年的具体几个月份，比如Jan,Mar,Jun。也可以使用连字符（-）指定连续的几个月，比如Jan-Jun，代表1月份到6月份。
- 对于某个月份的具体几个星期，只能输入一个值，比如FIRST或LAST。

表9-1 可用于Schtasks /Create的调度类型

调度类型	描 述	Modifier值
MINUTE	任务在指定的时间间隔（以分钟为计数单位）运行，默认情况下是1分钟运行一次	/mo 1~1439，任务两次运行之间间隔的分钟数，默认值为1
HOURLY	任务在指定的时间间隔（以小时为计数单位）运行，默认情况下是1小时运行一次	/mo 1~23，任务两次运行之间间隔的小时数，默认值为1

(续)

调度类型	描 述	Modifier值
DAILY	每天运行任务,或每隔n天运行任务,默认情况下是1天运行一次	/mo 1~365, 任务两次运行之间间隔的天数,默认值为1
WEEKLY	每星期或每隔n个星期的某天运行任务,默认情况下是每星期的周一运行一次	/mo 1~52, 任务两次运行之间间隔的星期数。可选地,还可以使用/d指定在某个星期的具体哪几天运行。MON、TUE、WED、THU、FRI、SAT、SUN分别指代周一到周日,*指代某星期中的每一天
MONTHLY	每个月或每隔n个月的某天运行任务,默认情况下是每个月的第一天运行任务	/mo 1~12, 任务两次运行之间间隔的星期数。可选地,还可以使用/d MON-SUN指定在某个月的星期几运行,*指代某月中的每一天
	另一种月份变量,用于指定某个月的特定日期,包括/mo与/m,或/m与/d	/mo LASTDAY,代表某个月的最后一天。/m JAN,FEB,...,DEC,设置具体哪一个月份。/d 1~31,设定某月的哪一天
	第三种月份变量,用于指定某个月的特定星期	/mo FIRST SECOND THIRD FOURTH LAST,设置某月中的哪个星期。/d MON-SUN,设置某星期的星期几。/m JAN,FEB,...,DEC,设置具体哪一个月份
ONCE	任务在指定的日期与时间运行一次	
ONEVENT	指定的事件发生或指定的事件日志中有事件发生时运行任务	/mo XPathString,其中,XPathString为XPath事件查询字符串,用于标识触发任务的事件
ONSTART	系统启动时运行任务	
ONLOGON	有用户登录时运行任务	
ONIDLE	系统空闲了指定的时间间隔后运行任务	/i 1~999,任务启动前系统空闲的分钟数

要进一步了解如何使用Schtasks /Create,参考下面给出的一些实例。

创建一个任务,该任务立即运行一次,之后不再运行:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc once
```

创建一个任务,该任务在系统启动时运行:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc onstart
```

创建一个任务,该任务在系统空闲时间超过10分钟后运行:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc  
onidle /i 10
```

创建一个任务,该任务在本地计算机上每隔15分钟运行:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc minute
```

```
/mo 15
```

创建一个任务，该任务在本地计算机上每隔5小时运行：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc hourly  
/mo 5
```

创建一个任务，该任务在本地计算机上每隔2天运行：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc daily  
/mo 2
```

创建一个任务，该任务在每隔两个星期的星期一（默认的运行日期）运行：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc weekly  
/mo 2
```

创建一个任务，该任务在每个星期的周一与周五运行：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc weekly  
/d mon, fri
```

创建一个任务，该任务在每个月第1天运行：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc monthly
```

创建一个任务，该任务在每隔一个月的第5天运行：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc monthly  
/mo 2 /d 5
```

创建一个任务，该任务在每个月的最后一天运行：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sysch.bat  
/sc monthly /mo lastday
```

创建一个任务，该任务在4月、8月、12月的第一个星期一运行：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sysch.bat /sc  
monthly /mo first /d mon /m apr,aug,dec
```

如果指定任务的文件路径中包含空格，要注意使用双引号对其进行封装，如下面实例所示：

```
schtasks /create /tn "SysChecks" /tr "c:\My Scripts\sch.bat" /sc onstart
```

如果没有使用双引号进行封装，则Schtasks尝试运行该任务时会出错。进一步地，如果需要向任务指定的程序、工具、脚本传递参数，可以将其附加在指定该任务的文件路径之后。如果参数中包含空格，也应该使用双引号对其进行封装，以便该参数被正确解释为单一的参数，而非多个参数。下面给出了几个实例：

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat 1 Y LAST  
/sc onstart
```



```
schtasks /create /tn "SysChecks" /tr "c:\My Scripts\sch.bat" Y N
/sc onstart
```

```
schtasks /create /tn "SysChecks" /tr "c:\My Scripts\sch.bat" "Full Checks"
```

你可以对远程计算机上的任务进行调度,也可以调度那些需要不同用户许可权限的任务。对远程计算机上的任务进行调度时,要记住的关键一点是当前使用的计算机必须与远程计算机在同一个域内,或者在远程计算机所信任的域内。为此,必须使用扩展的语法格式,包含如下的一些参数:

```
/s Computer /u [Domain\]User [/p Password]
```

其中, *Computer* 为远程计算机名或IP地址, *Domain* 为可选的域名, 用户账号就存在于该域内, *User* 为用户账号名(要使用的就是该用户账号的许可权限), *Password* 为该用户账号的口令(可选)。如果没有指定域, 则系统会假定当前域作为默认的域名。如果没有指定口令, 则会弹出提示信息要求输入口令。

要了解如何添加计算机与用户信息, 参考如下的实例。

在本地计算机上创建任务时, 使用账号 *adatum\wrstaneK*:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc onstart
/u adatum\wrstaneK /p RoverSays
```

创建任务时, 将远程计算机设置为 *mailer01*, 使用的账号为 *adatum\wrstaneK*:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc onstart
/s mailer01 /u adatum\wrstaneK /p RoverSays
```

通过 */Ru* 参数与 */Rp* 参数, 可以指定该任务运行时使用的用户账号的登录凭据。如果希望任务只有在特定用户登录时才运行, 可以使用可选的 */lt* 参数, 该参数规定任务应该以非交互方式运行, 并且只有在该任务的创建者登录时才运行。对于只需要操作本地资源的任务, 可以使用 */Np* 参数, 从而不需要为用户账号的登录凭据保存口令。要注意的是, 没有保存口令的情况下, 任务只能访问本地资源, 并以非交互方式由指定的用户运行。

默认情况下, 任务以标准用户权限运行。如果希望任务以指定用户的最高权限运行(比如要执行管理性任务), 可以将 */Rl* 参数设置为 *Highest*, 而不是默认的值 *Limited*。

要了解如何添加可替换的登录凭据与特权, 参考如下实例。

将任务配置为使用 *adatum\thomasv* 的登录凭据:

```
schtasks /create /tn "CleanUp" /tr c:\scripts\cleanup.bat /sc onlogon
/ru adatum\thomasv /rp DingoE
```

将远程计算机设置为 *server18*, 用于创建该任务的账号为 *adatum\wrstaneK*, 任务使用 *adatum\thomasv* 的登录凭据运行:

```
schtasks /create /tn "CleanUp" /tr c:\scripts\cleanup.bat /sc onlogon
/s server18 /u adatum\wrstaneK /p RoverSays /ru adatum\thomasv
/rp DingoE
```

使用账号 *adatum\wrstaneK* 在 *server18* 上运行任务, 不保存口令:

```
schtasks /create /tn "CleanUp" /tr c:\scripts\cleanup.bat /sc onlogon
```

```
/s server18 /u adatum\wrstanek /np
```

以最高权限运行任务:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc onlogon  
/rl highest
```

最后, 如果需要, 可以为任务添加特定的启动时间与日期, 以及结束时间与日期, 如下所示。

- */st StartTime*。其中, *StartTime* 为 24 小时制格式 (HH:MM), 比如, 15:00 代表下午 3 点。使用 */sc ONCE* 时, 这一参数是需要的。
- */et EndTime*。其中, *EndTime* 为 24 小时制格式 (HH:MM), 比如, 15:00 代表下午 3 点。调度类型为 *ONSTART*、*ONLOGON*、*ONIDLE*、*ONEVENT* 时, 这一参数是不适用的。
- */du Duration*。其中, *Duration* 为任务运行持续的小时数与分钟数, 其格式为 HHHH:MM。调度类型为 *ONSTART*、*ONLOGON*、*ONIDLE*、*ONEVENT* 时, 这一参数是不适用的。
- */sd StartDate*。其中, *StartDate* 为任务的开始日期, 采用默认的系统日期格式, 比如 MM/DD/YYYY。调度类型为 *ONCE*、*ONSTART*、*ONLOGON*、*ONIDLE*、*ONEVENT* 时, 这一参数是不适用的。
- */ed EndDate*。其中, *EndDate* 为任务的结束日期, 采用默认的系统日期格式, 比如 MM/DD/YYYY。调度类型为 *ONCE*、*ONSTART*、*ONLOGON*、*ONIDLE*、*ONEVENT* 时, 这一参数是不适用的。

提示 如果指定了终止日期或时间, 你也可以指定 */Z* 参数, 该参数的作用是完成任务的计划运行后删除该任务。

要了解如何指定特定的开始时间与日期, 以及终止时间与日期, 可以参考如下实例。
在午夜启动每小时执行一次的任务:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc hourly  
/st 00:00
```

在上午三点启动每小时执行一次的任务, 在上午七点结束:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc hourly  
/st 03:00 /et 07:00
```

在 2009 年 2 月 20 日上午三点启动每星期执行一次的任务:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc weekly  
/st 03:00 /sd 02/20/2009
```

在 2009 年 2 月 20 日上午三点启动每星期执行一次的任务, 在 2009 年 3 月 15 日上午 2:59 终止该任务:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc weekly  
/st 03:00 /sd 02/20/2009 /et 02:59 /ed 03/15/2009
```

注解 日期与时间格式是由区域与语言选项的设置决定的。在上面的这些实例中, 时区设置为 English (United States)。

9.3.2 创建由 Windows 事件触发的计划任务

通过 `Schtasks /Create`，可以创建操作系统或 Windows 组件向某个事件日志写入特定事件或特定类型事件时启动的任务。创建这种事件触发型任务的基本语法格式如下：

```
schtasks /create /tn TaskName /tr TaskToRun /sc ONEVENT
/ec LogName /MO EventString
```

其中，*TaskName* 为字符串形式的任务名，*TaskToRun* 指定了要运行的程序、命令行工具、脚本所在位置的文件路径，*LogName* 设置了要监控的事件日志名，*EventString* 设置了 XPath 事件查询字符串，该字符串用于识别触发任务的事件或事件集。

`/Sc ONEVENT` 的作用是将计划任务设置为由事件触发，`/Ec` 参数则指定了在命令行中可以使用的待监控事件日志名。如第8章8.1.2节所讲述，你可以使用 `Wevtutil el` 命令列出计算机上所有可用的事件日志。`/Mo` 参数用于指定 XPath 事件查询字符串，第8章中对 XPath 查询进行了深入讨论。

创建查询字符串时，并不是一定要完全重新创建。实际上，通过事件查看器创建过滤器，并由其识别待监控的事件或事件集是一个有效的方法。创建之后，参考第8章8.1.4节讲述的方法，将相关的 XPath 查询复制到记事本或其他文本编辑器中。复制到记事本中之后，保存该查询以便保留原始设置，之后从中提取必要的事件查询字符串。一般来说，需要的事件查询字符串存在于 Query 元素中的 `<Select>` 标签与 `</Select>` 标签之间。参考如下实例：

```
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[(Level=1 or Level=2 or Level=3)]]
  </Select>
  </Query>
</QueryList>
```

在这一实例中，事件查询字符串为：

```
*[System[(Level=1 or Level=2 or Level=3)]]
```

这一查询字符串创建了一个过滤器，其作用是在可用的事件日志中搜索关键性事件、警告事件与错误事件。下面的实例中，使用这一查询创建了一个名为 `Track Application Issues` 的计划任务，该任务的作用是在有关键性事件、警告事件与错误事件写入到可用的事件日志时运行事件查看器：

```
schtasks /create /tn "Track Application Issues" /tr wevtvwr.msc /
sc ONEVENT
/ec Application /MO "[System[(Level=1 or Level=2 or Level=3)]]"
```

要注意的是，由于任务名与查询字符串中包含空格，因此使用双引号对其进行封装。尽管这一查询字符串可能过于宽泛，但仍不失为一个了解查询字符串的参考实例。更有效的查询字符串用于识别特定事件（根据事件标识符），用于指定单独的事件标识符的事件查询字符串的语法格式如下：

```
* [System [ (EventID=EventNumber)]]
```

其中，*EventNumber* 为待监控事件的标识符。下面的实例中，创建了一个计划任务，使其在事件 ID 为 3210 的事件写入到系统日志时运行：

```
schtasks /create /tn "Computer Authentication Issues" /tr wevtvwr.msc
/sc ONEVENT/ec System /MO "[System[(EventID=3210)]]"
```

真实场景 计算机无法在域内进行认证时，事件ID为3210的事件就会写入到系统日志。在计算机口令需要重置时，就会产生这一事件，第14章14.3.4节对此进行了讨论。此外，如果计算机名与网络中另一台计算机重名，也会产生这一错误。

通过使用`or`语句对查询字符串进行扩展，可以输入多个事件标识符，扩展的语法格式如下：

```
*[System[ (EventID=EventNumber or EventID=EventNumber or ...)]]
```

下面的实例中，创建了一个计划任务，使其在事件ID为3210或5722的事件写入到系统日志时运行：

```
schtasks /create /tn "Computer Authentication Issues" /tr wevtvwr.msc  
/sc ONEVENT /ec System /MO "[System[(EventID=3210 or EventID=5722)]]"
```

真实场景 计算机被拒绝访问某资源时，事件ID为5722的事件就会写入到系统日志。在计算机账号被禁用或删除时就会产生这一错误。在第14章14.2.1与14.3.3两节中有关于这一主题更多的信息。

9.3.3 使用 Schtasks /Change 改变计划任务

你可以使用`Schtasks /Change`来改变与计划任务相关联的关键参数，其基本语法格式如下：

```
schtasks /change /tn TaskName ParametersToChange
```

其中，*TaskName*为想要改变参数的任务名，*ParametersToChange*为需要修改的参数，可以使用该命令操作的参数如下所示。

- **/ru Domain\User**。用于修改运行该任务的用户，比如，`/ru adatum\wrstaneek`。对系统账号，有效值包括 " "、" NT AUTHORITY\SYSTEM "、" SYSTEM "。对配置为用于操作Windows Vista与Windows Server 2008的任务，也可以使用 " NT AUTHORITY\ LOCAL- SERVICE "、" NT AUTHORITY\NETWORKSERVICE " 等账号。
- **/rp Password**。用于为以前指定的或新指定的用户账号设置口令。如果希望弹出输入口令的提示信息，此处可以使用 " * " 或空口令。对于系统账号，此处的口令是忽略的。只有在指定用户账号的同时，才可以为其设置口令。
- **/tr TaskToRun**。用于改变指定任务下运行的程序、命令行工具或脚本。
- **/st StartTime**。为每N分钟或每N小时运行的任务设置开始时间。其中，*StartTime*为24小时制格式 (HH:MM)，比如，15:00代表下午3点。在创建计划任务指定了/Sc ONCE时，这一参数是需要的。
- **/ri Interval**。设置了以分钟为计数单位的循环时间间隔，有效取值范围为1~599,940。计划类型为MINUTE、HOURLY、ONSTART、ONLOGON、ONIDLE、ONEVENT时，这一参数是不适用的。如果指定了/Et参数或/Du参数，则本参数值默认为10分钟。
- **/et EndTime**。为每N分钟或每N小时运行的任务设置结束时间。其中，*EndTime*为24小时制格式 (HH:MM)，比如，15:00代表下午3点。调度类型为ONSTART、ONLOGON、ONIDLE、ONEVENT时，这一参数是不适用的。
- **/du Duration**。设置了运行任务持续的小时数与分钟数，其格式为HHHH:MM。调度类型为ONSTART、ONLOGON、ONIDLE、ONEVENT时，或设置了/Et参数时，这一参数是不适用的。

- **/sd StartDate**。设置了任务的开始日期，采用默认的系统日期格式，比如MM/DD/YYYY。调度类型为ONCE、ONSTART、ONLOGON、ONIDLE、ONEVENT时，这一参数是不适用的。
- **/ed EndDate**。设置了任务的结束日期，采用默认的系统日期格式，比如MM/DD/YYYY。调度类型为ONCE、ONSTART、ONLOGON、ONIDLE、ONEVENT时，这一参数是不适用的。
- **/k**。当任务的结束时间或持续时间已到后，不应该再启动该任务。但如果任务已经在运行中，也不会强行终止该任务，而是等本次运行结束后不再启动。本参数生效的先决条件是必须已经指定了/Et参数或/Du参数。调度类型为ONSTART、ONLOGON、ONIDLE、ONEVENT时，这一参数是不适用的。
- **/it**。只有在任务所有者登录时才应该运行该任务。
- **/rl Level**。对标准用户权限，将任务运行级别设置为Limited。对runas用户的最高可能权限，将任务运行级别设置为Highest。
- **/delay DelayTime**。设置了任务触发到运行的时间延迟。DelayTime以mmmm:ss的形式进行设置，本参数只适用于ONSTART、ONLOGON、ONEVENT等调度类型。
- **/enable**。激活计划任务，使其可以根据调度计划运行。
- **/disable**。禁用计划任务，禁止其运行。
- **/z**。在任务最后一次调度运行后将其删除。

要了解如何改变任务，参考如下实例：

改变要运行的脚本：

```
schtasks /change /tn "SysChecks" /tr c:\scripts\systemchecks.bat
```

改变用户账号与口令：

```
schtasks /change /tn "SysChecks" /ru adatum\hthomas /rp gophers
```

将任务改变为在2009年3月1日上午七点开始每星期运行一次，在2009年3月30日上午6:59结束：

```
schtasks /change /tn "SysChecks" /st 07:00 /sd 03/01/2009 /et 06:59  
/ed 03/30/2009
```

注解 与前面类似，日期与时间格式是由区域与语言选项的设置决定的。在上面的这些实例中，时区设置为English (United States)。

对任务进行修改时，Schtasks会显示一条消息，并声明修改成功或失败，比如：

SUCCESS: The parameters of the scheduled task "SysChecks" have been changed.

如果操作的是远程计算机，或者当前登录的用户账号不具备修改任务的许可权限，则可以根据需要指定计算机与账号信息。其语法格式为：

```
schtasks /change /tn TaskName /s Computer /u [Domain\]User [/p Password]
```

其中，Computer为远程计算机名或IP地址，Domain为可选的域名，用户账号就存在于该域内，User为用户账号名（要使用的就是该用户账号的许可权限），Password为该用户账号的口令（可选）。如果没有指定域，则系统会假定当前域作为默认的域名。如果没有指定口令，则会弹出提示信息要求输入口令。

下面的实例中，远程计算机为mailer01，有权限修改任务SysChecks的用户账号为adatum\wrstanek：

```
schtasks /change /tn "SysChecks" /tr c:\scripts\systemchecks.bat
/s mailer01 /u adatum\wrstanek
```

由于上面命令中没有指定口令，因此，执行时Schtasks会要求输入口令。

你可以根据任务名快速地对其进行激活或禁用，激活任务的语法格式如下：

```
schtasks /change /tn TaskName /enable
```

禁用任务的语法格式如下：

```
schtasks /change /tn TaskName /disable
```

其中，*TaskName*为要激活或禁用的任务名，比如：

```
schtasks /change /tn "Syschecks" /disable
```

9.3.4 使用 Schtasks/Query 查询已配置的任务

通过在命令提示符中键入schtasks/query命令，可以快速地确定计算机上配置了哪些任务。对远程计算机，可以指定必要的计算机与账号信息，使用如下的语法格式：

```
schtasks /query /s Computer /u [Domain\]User [/p Password]
```

其中，*Computer*为远程计算机名或IP地址，*Domain*为可选的域名，用户账号就存在于该域内，*User*为用户账号名，该账号具备远程计算机上适当的访问许可权限，*Password*为该用户账号的口令（可选）。

下面的实例中，远程计算机为mailer01，用户账号为adatum\wrstanek：

```
schtasks /query /s mailer01 /u adatum\wrstanek
```

由于上面命令中没有指定口令，因此，执行时Schtasks会要求输入口令。

Schtasks/Query命令的基本输出信息以表格形式呈现，包括TaskName、Next Run Time、Status等列。通过分别使用/Fo List或/Fo Csv参数，也可以使得输出信息分别呈现为列表形式与逗号分隔的多行形式。使用列表格式输出时，最好同时使用/V（verbose）参数，该参数提供了所有任务属性的完整的详细资料，如下面实例所示：

```
schtasks /query /s mailer01 /u adatum\wrstanek /fo list /v
```

另一个有用的参数是/Nh，该参数规定表格形式或CSV格式的输出中不带头信息。

提示 你可能会疑惑为什么要使用不同的输出格式，实际上不同的输出格式可以满足不同的需求。如果需要查看系统中已配置所有任务的详细资料，或者正在进行故障排除，建议使用详细列表格式（/Fo List/V）。如果需要将输出信息存储到一个文件中，并且日后可能会将其导出到电子表格或非关系型数据库，建议使用逗号分隔的形式。另外，也可以使用输出重定向符号（>或>>）将Schtasks的输出信息进行重定向。

9.3.5 使用 XML 配置文件创建任务

在Schtasks/ Create命令的讲解中，没有讨论/Xml参数。在使用Schtasks/ Create命令时，通过/Xml参数，可以指定XML配置文件，其中定义了要创建的任务。其基本语法格式如下：

```
schtasks /create /tn TaskName /xml XmlFile
```

其中, *TaskName*为要创建的任务名, 指定了XML配置文件名或全文件路径, XML配置文件中包含了任务的设置信息, 比如:

```
schtasks /create /tn "Housekeeping Task" /xml housekeepingtask.xml
```

与其他方式创建任务类似, 这里也可以使用/S参数来指定远程计算机, /U参数指定创建该任务的用户的上下文, /P参数指定用户口令。尽管任务的XML配置文件中可以定义替换的登录凭据(用于运行该任务), 但也可以使用/Ru参数与/Rp参数指定替换的登录凭据。

在XML配置文件中, 一般不直接指定实际的用户口令, 而是将其设置为空或*, 之后在使用/Rp参数创建任务时指定必要的口令。

下面给出了一个XML配置文件实例, 其中定义了一个计划任务及其设置信息:

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2009/02/
mit/task">
  <RegistrationInfo>
    <Date>2009-10-01T18:10:12</Date>
    <Author>WilliamS</Author>
  </RegistrationInfo>
  <Triggers>
    <EventTrigger>
      <StartBoundary>2009-10-01T18:10:00</StartBoundary>
      <Enabled>true</Enabled>
      <Subscription>&lt;QueryList&gt;&lt;Query&gt;&lt;Select Path='system'
&gt;* [System [ (Level=1 or Level=2 or Level=3)]
&lt;/Select&gt;&lt;/Query&gt;&lt;/QueryList&gt;</Subscription>
    </EventTrigger>
  </Triggers>
  <Settings>
    <IdleSettings>
      <Duration>PT10M</Duration>
      <WaitTimeout>PT1H</WaitTimeout>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
```

```

    <Command>wevtvwr.msc</Command>
  </Exec>
</Actions>
<Principals>
  <Principal id="Author">
    <UserId>ADATUM\WILLIAMS</UserId>
    <LogonType>InteractiveToken</LogonType>
  </Principal>
</Principals>
</Task>

```

从上面的文件可以看出，XML配置文件是相当复杂的。当然，并不是必须从头开始创建XML配置文件，通过9.2.5节中的讲述，你可以将现存任务的设置信息导出到XML配置文件中，之后使用这一文件在其他计算机上创建同样的任务。

在命令行中，可以使用Schtasks/Query命令来显示任务的状态及其XML配置文件。其中，/Tn参数后面跟随要操作的任务名，/Xml参数用于显示XML配置文件及任务状态。下面的实例显示了任务Computer Authentication的状态及其配置文件：

```
schtasks /query /tn "Computer Authentication" /xml
```

如果将上面命令的输出重定向到.xml文件，之后对其进行编辑，去除其中状态相关的详细资料，就获得了一个完整的XML配置文件，可以用于创建任务。下面的实例中，任务Computer Authentication的配置信息被写入到一个名为ComputerAuthTask.xml的文件中：

```
schtasks /query /tn "Computer Authentication" /xml > ComputerAuthTask.xml
```

操作远程计算机时，可以使用/S参数指定远程计算机，/U参数指定创建任务的用户的上下文，/P参数指定用户口令。

了解了如何操作XML配置文件后，就可以在需要的时候人工编辑XML配置文件。进行编辑时，要记住应该先在测试或开发系统上对所做的修改进行测试，而不要直接在运营系统上进行操作。对于XML配置文件，要注意如下讲述的一些要点。

RegistrationInfo元素指定了任务的创建时间与创建者：

```

<RegistrationInfo>
  <Date>2009-10-01T18:10:12</Date>
  <Author>WilliamS</Author>
</RegistrationInfo>

```

Triggers元素指定了任务运行的条件，这一元素包含了如下的一些元素。

- EventTrigger元素。用于定义由特定事件触发的任务。
- TimeTrigger元素。用于定义由时间触发的一次性任务或周期执行的任务。
- BootTrigger元素。用于定义系统启动时触发的任务。
- IdleTrigger元素。用于定义计算机处理资源空闲时触发的任务。
- RegistrationTrigger元素。用于定义在任务被创建或修改时触发的任务。

Boot、idle以及registration是最易于定义的触发类型，因为这些类型的任务或者是激活的，或者是禁用的，如下面实例所示：

```

<Triggers>
  <BootTrigger>
    <Enabled>true</Enabled>

```

```

    </BootTrigger>
  </Triggers>

```

Actions元素用于定义待运行的命令、待发送的电子邮件，或者待显示的消息等。待运行的命令是在Exec元素内定义的。下面的实例中，运行了一个名为CleanUp.bat的脚本：

```

<Actions Context="Author">
  <Exec>
    <Command>c:\scripts\cleanup.bat</Command>
  </Exec>
</Actions>

```

待发送的电子邮件是在SendEmail元素中定义的。下面的实例定义了一条经由邮件服务器mailer15.adatum.com发送给admins@adatum.com的电子邮件：

```

<Actions Context="Author">
  <SendEmail>
    <Server>mailer15.adatum.com</Server>
    <Subject>Possible Database Outage</Subject>
    <To>admins@adatum.com</To>
    <From>williams@adatum.com</From>
    <Body>The CRM Database appears to be down.</Body>
    <HeaderFields/>
  </SendEmail>
</Actions>

```

显示在桌面的消息是在ShowMessage元素中定义的。下面的实例显示了一条告警消息，提示了可能的应用程序中断：

```

<Actions Context="Author">
  <ShowMessage>
    <Title>Application Outage Warning</Title>
    <Body>The CRMComms application is having write errors.</Body>
  </ShowMessage>
</Actions>

```

最后，Principals元素定义了运行该任务的用户的上下文，包括该任务是否以非交互模式运行、运行级别等。下面的实例中，任务以用户账号WilliamS的最低权限非交互地运行：

```

<Principals>
  <Principal id="Author">
    <UserId>CPANDL\williams</UserId>
    <LogonType>InteractiveToken</LogonType>
    <RunLevel>LeastPrivilege</RunLevel>
  </Principal>
</Principals>

```

你可以将任务配置为根据用户是否登录运行，其方法是将LogonType设置为Password。也可以将任务配置为以最高权限运行，方法是将RunLevel设置为HighestAvailable。下面给出了应用这些选项的一个实例：

```

<Principals>
  <Principal id="Author">
    <UserId>CPANDL\williams</UserId>
    <LogonType>Password</LogonType>

```

```
<RunLevel>HighestAvailable</RunLevel>
</Principal>
</Principals>
```

如果不希望任务计划程序为用户账号存储口令，可以将LogonType设置为S4U，如下面实例所示：

```
<Principals>
  <Principal id="Author">
    <UserId>CPANDL\williams</UserId>
    <LogonType>S4U</LogonType>
    <RunLevel>HighestAvailable</RunLevel>
  </Principal>
</Principals>
```

9.3.6 使用 Schtasks /Run 立即运行任务

通过如下的语法格式，可以在任意时刻立即运行任务：

```
schtasks /run /tn TaskName
```

其中，TaskName为所要运行的任务名，比如：

```
schtasks /run /tn "SysChecks"
```

使用schtasks /run运行命令并不会影响该任务的调度计划，也不会影响该任务下一次运行的时间。如果该任务可以成功启动，你会看到一条消息说明这一点。此外，你可以指定远程计算机名（任务在其上配置）。必要的时候，还可以指定运行该任务的账号以及可选的口令，如下面实例所示：

```
schtasks /run /tn "SysChecks" /s 192.168.1.100
schtasks /run /tn "SysChecks" /s 192.168.1.100 /u adatum\wrstaneK
```

注解 如果指定用户账号但没提供口令，则会立即看到提示信息要求输入口令。

9.3.7 使用 Schtasks /End 终止运行中的任务

通过如下的语法格式，可以在任意时刻终止任务：

```
schtasks /end /tn TaskName
```

其中，TaskName为正在运行的、需要终止的任务名，比如：

```
schtasks /end /tn "SysChecks"
```

如果该任务本来正处于运行状态，则执行上面命令后，该任务被终止。成功执行后，会看到类似于如下的输出信息：

```
SUCCESS" The scheduled task "SysChecks" has been terminated successfully.
```

你可以指定远程计算机名（任务在其上配置），必要的时候，还可以指定有权限终止该任务的账号以及可选的口令，如下面实例所示：

```
schtasks /end /tn "SysChecks" /s 192.168.1.100
```

或：

```
schtasks /end /tn "SysChecks" /s 192.168.1.100 /u adatum\wrstaneK
```


由于没有指定口令，Schtasks会弹出提示信息要求输入口令。

9.3.8 使用 Schtasks/Delete 删除任务

你可以根据任务名删除本地或远程计算机上的任务，采用如下的语法格式：

```
schtasks /delete /tn TaskName [/s Computer /u [Domain/]User [/p Password]]
```

其中，*TaskName*为待删除的任务名，其后的参数是可选的，用于标识远程计算机、删除任务时使用的用户账号以及该账号的口令，比如：

```
schtasks /delete /tn "SysChecks"
```

或：

```
schtasks /delete /tn "SysChecks" /s 192.168.1.100 /u adatum\wrstanek  
/p frut5
```

注解 如果指定用户账号但没提供口令，则会立即看到提示信息要求输入口令。

输入Schtasks /Delete命令之后，会看到一条警告消息，询问是否确认删除该任务，此时根据需要键入适当的字母即可。如果不希望看到提示信息，可以使用/F参数，比如：

```
schtasks /delete /tn "SysChecks" /f
```

通过上面的命令，就可以直接删除该任务，而不会看到警告信息。

此外，如果需要删除本地计算机或指定的远程计算机上的所有任务，可以输入*作为任务名，比如：

```
schtasks /delete /tn *
```

弹出提示信息时，确认。



Part 3

第三部分

使用命令行管理 Windows 文件系统和磁盘

本 部 分 内 容

- 第 10 章 配置与维护磁盘
- 第 11 章 对基本磁盘进行分区
- 第 12 章 管理动态磁盘上的卷与 RAID

本章将讲述关于磁盘配置与维护的技术，实际上这里涉及的内容比大多数人想象的要多得多。Windows Server 2008与Windows Vista支持硬盘驱动器与可移动存储设备。硬盘驱动器可以配置为基本的和动态的两种磁盘类型，以及主引导记录（MBR）与GUID分区表（GPT）两种磁盘分区类型；可移动存储设备的磁盘类型为可移动的。

10.1 使用 DiskPart

要操作磁盘、磁盘分区和卷，DiskPart是首选的工具，它可以完成的主要任务包括转换磁盘类型、创建磁盘分区与卷、配置RAID等。此外，可以使用DiskPart对新磁盘的自动挂载进行配置，也可以通过它分配驱动器盘符与驱动器路径，还可以用它格式化磁盘。要格式化磁盘，可以使用FORMAT命令，该命令将在11.4节讨论。

10.1.1 DiskPart 基础

与本书中前面讲述的所有其他命令不同的是，DiskPart并不是一个可以使用命令行与参数调用的简单的命令行工具，而是一个文本模式的可调用的命令解释器，这样你可以使用一个单独的命令提示符与一些内部命令来管理磁盘、磁盘分区和卷。要调用DiskPart命令解释器，可以在命令窗口中键入diskpart，之后按Enter键。

DiskPart的作用是操作计算机上安装的物理硬盘，可以是内部的、外部的，也可以是混合的。尽管DiskPart也会列出其他类型的磁盘，比如CD/DVD驱动器、可移动存储介质以及连接在USB接口的闪存设备，也可以执行一些基本的任务，比如分配盘符，但DiskPart不能完成对这些设备的完全管理与操作。

在使用DiskPart命令之前，必须首先列出并选定待操作的磁盘、分区或卷。选定某个具体的磁盘、分区或卷之后，键入的DiskPart命令就会作用于该设备。

通过如下的list命令，可以分别列出系统中的磁盘、卷与分区。

- ❑ list disk。列出计算机上安装的所有物理硬盘。
- ❑ list volume。列出计算机上安装的所有卷（包括硬盘分区与逻辑驱动器）。
- ❑ list partition。列出选定的磁盘上的分区。

使用这些list命令后，星号会出现在选定的磁盘、卷、分区之后。磁盘或分区的选择是通过其编号实现的，卷的选择是通过其编号或盘符实现的，比如disk 0、partition 1、volume 2、volume D等。

使用DiskPart解释器完成操作后，键入exit，就可以退出DiskPart提示符，并返回到标准的命令行。

10.1.2 DiskPart: 一个实例

要了解如何使用DiskPart, 可以参考下面的实例, 该实例调用了DiskPart, 列出了可用的磁盘, 并将焦点放置到disk 2。

(1) 在命令提示符中键入**diskpart**, 调用DiskPart。

(2) 执行后, 命令提示符切换为

DISKPART>

(3) 此时处于文本模式的DiskPart解释器中。要列出可用的磁盘, 可以在命令提示符中键入**list disk**。

(4) **list disk**的输出信息列出了计算机中可用的磁盘及其状态、大小与可用空间:

Disk ###	Status	Size	Free	Dyn	Gpt
-----	-----	-----	-----	---	---
Disk 0	Online	466 GB	1528 KB		
Disk 1	Online	466 GB	1528 KB		
Disk 2	Online	233 GB	0 B	*	

(5) 由于disk 2是需要操作的磁盘, 因此, 键入**select disk 2**命令, 将焦点放置在其上。

(6) DiskPart报告:

“磁盘2现在是所选磁盘。”

(7) 根据需要对该磁盘进行操作, 完成后, 键入**exit**, 就可以退出DiskPart解释器。

10.1.3 理解焦点及其内涵

选定了一个磁盘、分区或卷后, 焦点自动放置在该设备, 直到选择了另外的设备。前面的实例中, 焦点放置在disk 2, 但如果选择disk 0上的volume 2, 则焦点会从disk 2转移到disk 0, volume 2。有些情况下, 焦点会根据使用的命令自动变换。比如, 创建分区或卷后, 焦点会自动切换到新的分区或卷。

要注意的是, 只能将焦点放置到当前选定磁盘上的某个分区。当焦点转移到某个分区上之后, 其相关的卷也相当于放置了焦点。卷放置了焦点之后, 相关的磁盘与分区也相当于放置了焦点 (如果该卷映射到单一的特定分区)。如果该卷没有映射到单一的特定分区, 则只有该卷具备焦点。

10.1.4 DiskPart 命令与脚本

LIST命令与SELECT命令只是DiskPart命令提供的众多命令中的两个。表10-1展示了DiskPart命令的完整列表, 其中的很多命令接受*Noerr*作为另外的参数。该参数可用于DiskPart脚本中, 其作用是在发生错误时, 告诉DiskPart继续处理脚本中的命令。如果没有指定*Noerr*参数, 则发生错误时, DiskPart命令会退出并返回错误代码, 使得脚本的执行中断。

- 可以使用*Noerr*参数并在退出时返回错误代码的命令包括ADD、ASSIGN、ATTRIBUTES、AUTOMOUNT、BREAK、CONVERT、CREATE、DELETE、EXTEND、IMPORT、OFFLINE、ONLINE、RECOVER、REMOVE、REPAIR、SAN、SETID、SHRINK、UNIQUEID。
- 不使用*Noerr*参数或者在退出时返回错误代码的命令包括ACTIVE、CLEAN、DETAIL、EXIT、FILESYSTEMS、GPT、HELP、INACTIVE、LIST、REM、RESCAN、RETAIN、SELECT。

表10-1 DiskPart命令总结

命 令	描 述	语 法
ACTIVE	在MBR磁盘上, 将当前焦点所在的分区标记为活跃的系统分区, 意味着该分区包含了操作系统启动文件	active
ADD	在选定的动态磁盘上创建一个镜像卷	add disk= <i>n</i> , 其中 <i>n</i> 为包含镜像的磁盘编号 add disk= <i>n</i> [align= <i>nn</i>]
ASSIGN	为选定的分区、逻辑驱动器或卷分配盘符或挂载点	assign letter= <i>x</i> assign mount= <i>path</i>
ATTRIBUTES	显示或管理磁盘或卷上的属性	attributes disk [set clear] [readonly] (注意SP1版本前的Windows Vista不支持disk子命令) attributes volume [set clear] [hidden readonly nodefaultdriveletter shadowcopy] (注意SP1版本前的Windows Vista不支持nodefaultdriveletter与shadowcopy子命令)
AUTOMOUNT	用于控制是否自动挂载新加入到系统中的基本卷并为其分配盘符	automount[enable disable scrub]
BREAK	中断一个镜像集。添加nokeep, 以便规定只保留一个卷, 这意味着删除其他卷(只适用于Windows Server 2008)	break disk= <i>n</i> break disk= <i>n</i> nokeep
CLEAN	移除焦点所在磁盘上所有格式化的分区或卷, 使用CLEAN ALL时, 所有磁盘扇区被清零	clean clean[all]
CONVERT	在不同的磁盘格式间转换	convert basic dynamic convert gpt mbr
CREATE	创建特定类型的分区或卷	create partition efi extended logical msr primary create volume simple raid stripe
DELETE	删除焦点所在的磁盘、分区或卷	delete disk partition volume
DETAIL	提供焦点所在磁盘、分区或卷的详细资料	detail disk partition volume
EXIT	退出DiskPart解释器	exit
EXTEND	扩展选定磁盘上的简单卷, 或者跨越多个磁盘上的简单卷	extend size= <i>n</i> disk= <i>n</i> extend filesystem
FILESYSTEMS	显示卷上当前的与支持的文件系统类型	filesystems
FORMAT	对选定的卷进行格式化	format [[format fs= <i>type</i>][revision]][recommended] [label= <i>label</i>][unit= <i>n</i>][quick][compress][override] [nowait]
GPT	改变焦点所在分区的GPT属性	gpt attributes= <i>n</i> , 其中, <i>n</i> 为包含16个字符的16进制数值
HELP	为指定的命令显示支持的命令列表或帮助信息	help help <i>command name</i>

(续)

命 令	描 述	语 法
IMPORT	导入外部磁盘	import
INACTIVE	在MBR磁盘上, 将当前焦点所在的分区标记为不活跃的分区, 意味着计算机不会从该系统分区启动, 而是从固件中选择下一个引导项	inactive
LIST	显示磁盘或卷的列表及其相关信息, 或者焦点所在磁盘上的分区列表	list disk partition volume
ONLINE	将选定的磁盘或卷联机对焦点所在的镜像(或RAID-5)卷进行重新同步	online disk online volume
OFFLINE	将选定的磁盘脱机(只适用于Windows Server 2008)	offline disk
RECOVER	尝试对与选定的动态磁盘相关联的RAID-5卷进行恢复与重新同步(只适用于Windows Server 2008)	recover
REM	在DiskPart脚本中标记注释的开始	rem <i>comment</i>
REMOVE	从当前选定的卷上移除盘符或挂接点。可选的是, 可以添加Dismount参数	remove letter= <i>x</i> remove mount= <i>path</i> remove all
REPAIR	修复焦点所在的RAID-5卷, 其方法是使用指定的动态磁盘替换该卷(只适用于Windows Server 2008)	repair disk= <i>n</i> [align= <i>nn</i>]
RESCAN	搜索可能已经添加到计算机上的新磁盘	rescan
RETAIN	将选定的简单卷预备用于引导卷或系统卷	retain
SAN	为当前引导的操作系统显示或设置存储区域网络(SAN)策略(Windows Vista SP1及后续版本或者Windows Server 2008)	san san policy= <i>value</i>
SELECT	选定磁盘、分区或卷, 并将焦点放置到其上	select disk partition volume
SETID	在选定的分区上设置分区类型	set id= <i>value</i> [override]
SHRINK	降低选定卷的容量, 其方法是, 将空闲空间转变为卷尾部的未使用空间	shrink [desired= <i>n</i>][nowait] shrink [minimum= <i>n</i>][nowait] shrink querymax
UNIQUEID	为磁盘显示或设置GPT标识符或MBR签名(Windows Vista SP1及后续版本或者Windows Server 2008)	Uniqueid disk[id= <i>value</i>]

说到DiskPart脚本, 其使用方法与其他命令脚本有些差别。因为DiskPart是一个文本模式的解释器, 而不是一个标准的命令行工具。调用DiskPart时(方法是在命令提示符中键入**diskpart**), 要附加/S参数, 以便DiskPart解释器获知要使用的脚本, 如下所示:

```
diskpart /s ScriptName.txt
```

其中, *ScriptName.txt*为文本文件名, 其中包含了要使用的脚本。默认情况下, DiskPart的输出写入到当前的命令提示符, 但也可以对其进行重定向, 如下所示:

```
diskpart /s ScriptName.txt > LogFile.log
```

或

```
diskpart /s ScriptName.txt >> LogFile.log
```

其中, *LogFile.log* 是 DiskPart 输出将要写入的文件名。

注解 重定向符号 > 用于创建或重写文件, >> 用于创建文件或向现存文件添加内容。

提示 使用脚本而不是直接输入命令的好处是: 可以将磁盘相关任务自动化, 以便以完全相同的方式重复执行。在采用无人值守的安装模式部署 Windows 系统时, 这种脚本化的磁盘管理任务也是有益的。

使用 DiskPart 脚本时, 要注意如下的一些错误代码。

- 0。表示没有错误发生, 执行顺利。
- 1。发生致命意外, 可能存在严重问题。
- 2。表示为命令指定的参数是错误的。
- 3。表示 DiskPart 无法打开指定的脚本或输出文件。
- 4。表示 DiskPart 使用的服务返回错误代码或报告错误。
- 5。表示发生了命令语法错误, 典型的原因是错误地选择了磁盘、分区或卷, 或者不适用于该命令。

10.1.5 DiskPart: 脚本实例

使用 DiskPart 脚本时, 应该将所要执行的所有操作作为某一次会话的一部分。脚本中应该包含所有待执行的 DiskPart 命令, 但没有必要包含 EXIT 命令, 因为文本模式的解释器会在脚本执行到结尾时自动退出。命令清单 10-1 给出了一个 DiskPart 脚本实例:

命令清单 10-1 DiskPart: 脚本实例

```
rem Select disk 2
select disk 2

rem Create the primary partition on the disk and assign the drive letter
create partition primary size=4096 assign letter=s
format fs=ntfs label="primary"
rem Create extended partition with 2 logical drives
create partition extended size=4096
create partition logical size=2048
assign letter=u
format fs=ntfs label="extended1" create partition logical size=2047
assign letter=v format fs=ntfs label="extended2"
```

上面的脚本实例中, 在 disk 2 上创建了主分区与扩展分区。主分区大小设置为 4096MB, 盘符设置为 S, 并使用 NTFS 文件系统进行格式化。扩展分区大小为 4096MB, 分为两个逻辑分区。第一个逻辑分区大小设置为 2048MB, 盘符为 U, 使用 NTFS 文件系统进行格式化; 第二个逻辑分区大小设置为 2048MB, 盘符为 V, 使用 NTFS 文件系统进行格式化。这样设置逻辑分区大小的原因是分区过程会损

失一些磁盘空间，如果不需要分为两个逻辑分区，也可以只创建一个大小为4096MB的逻辑分区。

注解 按上面实例中这样创建分区并分配盘符之后，分区并不能直接使用，仍然需要使用DiskPart中的FORMAT命令对其进行格式化。要获取关于格式化分区与卷的更多信息，可以参考第11章11.4节。

提示 由于DiskPart命令对磁盘的变更到实际生效有一个过程，因此，不应该连续运行多个DiskPart脚本，而是应该在运行多个脚本之间（或处理单一DiskPart会话的所有任务之间）间歇10~15秒。这样做不仅可以保证前一次DiskPart会话发布的最后一条命令执行完毕，还可以保证前一次DiskPart会话在下一次会话开始之前完成并关闭。

要运行DiskPart脚本，可以键入**diskpart /s ScriptName**，比如**diskpart /s disk2config.txt**。运行该脚本时，应该有类似于如下的输出信息：

```
Disk 2 is now the selected disk.
DiskPart succeeded in creating the specified partition.
DiskPart successfully assigned the drive letter or mount point.
DiskPart successfully formatted the volume.
DiskPart succeeded in creating the specified partition.
DiskPart succeeded in creating the specified partition.
DiskPart successfully assigned the drive letter or mount point.
DiskPart successfully formatted the volume.
DiskPart succeeded in creating the specified partition.
DiskPart successfully assigned the drive letter or mount point.
DiskPart successfully formatted the volume.
```

从输出信息可以看出，对脚本运行的每一步骤，DiskPart都报告了成功或失败的结果。需要指出的是，DiskPart脚本也可以不保存在本地计算机上。如果DiskPart脚本保存在网络共享位置\\corpserver01\scripts处，则可以通过如下命令调用该脚本：

```
diskpart/s \\corpserver01\scripts\disk2config.txt
```

上面命令可以成功执行的前提是从本地系统可以访问该网络共享位置。为确保这一点，也可以使用NET USE命令来映射网络驱动器，其格式为：

```
net use DriveLetter: \\ComputerName\ShareName
```

比如，对上面的实例，可以使用如下命令来映射网络驱动器：

```
net use X: \\corpserver01\scripts
```

注解 使用NET USE命令时，可以以/USER:Domain\User的格式提供用户名与口令信息。也可以指定映射的驱动器是否是持久化的，也就是说，计算机重启后，网络共享映射是否仍然有效。其方法是使用参数/Persistent:Yes。如果需要删除持久化的网络共享，则可以键入**net use \\ComputerName\ShareName/DELETE**命令。

默认情况下，如果DiskPart脚本在执行命令时遇到错误，就会停止执行该脚本，并返回一个错误

代码。但如果为其中的每一条命令都指定`Noerr`参数,则DiskPart脚本在执行命令出错后也会报错,但会继续执行后续命令。此外,并不是必须在命令行中直接键入命令,命令可以是大型脚本(可以称为master脚本)的一部分,命令清单10-2给出了一个master脚本实例。

命令清单10-2 master脚本实例

```
@echo off
@if not "%OS%"=="Windows_NT" goto :EXIT
@if "%1"==" " (set INFO=echo && set SEXIT=1) else (set INFO=rem && set SEXIT=0)

%INFO% *****
%INFO% Script: Disk2Setup.bat
%INFO% Creation Date: 3/3/2008
%INFO% Last Modified: 3/15/2008
%INFO% Author: William R. Stanek
%INFO% Email: williamstanek@aol.com
%INFO% *****
%INFO% Description: Configures the standard partitions on workstations
%INFO%                  with a third hard drive. The script is configured so
%INFO%                  that it will only run if you pass in a parameter,
%INFO%                  which can be any value. This is meant as a
%INFO%                  safeguard to help prevent accidental formatting
%INFO%                  of disks.
%INFO% *****
@if "%SEXIT%"=="1" goto :EXIT

@title "Configuring Disk 2..."
cls
color 07

net use x: \\corpserver01\scripts
diskpart /s x:\disk2config.txt

rem perform other necessary tasks here
:EXIT
echo Exiting...
```

本章前面对DiskPart命令进行了一些介绍,下面将讨论如何使用DiskPart命令及其相关命令(比如CHKDSK与DEFRAG)来创建、管理、维护磁盘、分区与卷。

10.2 安装与管理硬盘驱动器

使用DiskPart的一个重要原因是有助于配置与维护硬盘驱动器。关键的管理任务包括检查新驱动器、确定驱动器状态以及管理分区表风格等。

10.2.1 安装与检查新驱动器

Windows操作系统同时支持热插拔驱动器与非热插拔驱动器。热插拔是一种有益的功能,使得用户可以在不关机的情况下移除驱动器。大多数情况下,热插拔驱动器都是从计算机正面安装与移除的,如果某台计算机支持热插拔,你就可以在不关机的情况下安装驱动器。进行驱动器的热插拔后,可以

键入**rescan**命令来寻找新设备。新驱动器是以适当类型（基本的或动态的）的磁盘的形式添加到系统中的。如果没有找到新添加的驱动器，则可能需要重新引导计算机。

如果计算机不支持驱动器的热插拔，则安装驱动器时必须先关机。安装后，如果需要，则可以用前面描述的方法来搜索新添加的驱动器。

10.2.2 检查驱动器状态与配置

通过在DiskPart提示符中键入**list disk**命令，可以检查驱动器的状态。该命令典型的输出信息类似于如下格式：

```

Disk ###      Status      Size      Free      Dyn      Gpt
-----
Disk 0        Online      466 GB    1528 KB
Disk 1        Online      466 GB    1528 KB
Disk 2        Offline     233 GB    230 GB    *

```

从上面的输出信息可以看出，**list disk**命令展示了系统中已配置磁盘如下一些要素。

- **Disk ###**。磁盘编号。
- **Status**。磁盘的当前状态。
- **Size**。磁盘的总容量。
- **Free**。可用于分区的磁盘空间（不是磁盘上实际可用空间的总量）。
- **Dyn**。如果本列中是*，则说明该磁盘是动态磁盘，否则是基本磁盘。
- **Gpt**。如果本列中是*，则说明该磁盘分区表类型是GPT，否则是MBR。

前面的实例中，输出信息表明该计算机上有3块使用MBR分区类型的基本磁盘。其中，disk 0与disk 1状态为联机，disk 2状态为脱机，但通过将焦点切换到disk 2（键入**select disk 2**，之后键入**online**），可以让disk 2状态由脱机变为联机。

可以看出，在安装新驱动器或排除驱动器问题时，获知驱动器状态是有用的，表10-2总结了常用的驱动器状态值。

表10-2 常用驱动器状态值及其含义

状 态	描 述	含 义
音频CD (Audio CD)	CD/DVD驱动器中放置了一个音频CD	驱动器一切正常
外部 (Foreign)	动态磁盘已移动到计算机中，但尚未导入并使用。有时候，失效的驱动器重新联机后也可能标记为外部磁盘	需要使用IMPORT命令将磁盘添加到系统中
初始化 (Initializing)	将基本磁盘转换为动态磁盘时经历的一个临时状态	初始化完成时，状态应该自动变化为联机
丢失 (Missing)	动态磁盘损坏、关闭或失去连接，该值出现在磁盘标识符中，而非状态列	重新连接或打开失去连接的磁盘，之后使用RESCAN命令来定位卷。如果不再使用该磁盘，可以使用DELETE DISK命令从磁盘列表中删除该磁盘
没有介质 (No Media)	CD-ROM或可移动驱动器中尚未放置存储介质，只有CD-ROM或可移动磁盘类型才会显示这一状态	插入CD-ROM、软盘或可移动存储介质，以便使得磁盘状态变为联机

(续)

状 态	描 述	含 义
尚未初始化 (Not Initialized)	磁盘不包含有效的签名。在通过新磁盘检测向导第一次启动磁盘管理时, Windows会为磁盘赋予MBR或GPT类型。如果取消了该向导, 就会出现这一状态	如果尚未启动磁盘管理, 启动它, 之后使用初始化磁盘向导写入磁盘签名。否则, 在磁盘管理中右击某磁盘, 之后选择初始化磁盘
脱机 (offline)	动态磁盘不可访问, 可能是因为损坏或暂时不可用。如果磁盘名变为丢失, 则系统可能无法再定位或识别该磁盘	检查驱动器及其控制器与缆线的问题, 确认驱动器电路及线路是否正确连接。如果可能的话, 使用ONLINE命令将磁盘状态变为联机
联机 (Online)	正常的磁盘状态, 表示磁盘可以访问, 不存在问题。基本磁盘与动态磁盘都可以显示这一状态信息	磁盘不存在任何已知的问题
联机 (错误) [Online (Errors)]	动态磁盘中检测到了输出/输出 (I/O) 错误	你可以尝试使用ONLINE命令纠正临时性的错误, 也可以使用RECOVER命令重新同步镜像卷或RAID-5卷
不可读 (Unreadable)	磁盘当前不可访问, 重扫描磁盘时可能发生这一情况。基本磁盘与动态磁盘都可能显示这一状态信息	如果驱动器不是正处于扫描状态, 则该驱动器很可能已损坏或存在I/O错误。可以使用RESCAN命令重扫描磁盘并进行读取操作 (如果可能), 也可以尝试重新引导系统
未识别 (Unrecognized)	磁盘类型未知, 无法在系统中使用, 来自非Windows系统的磁盘在Windows中可能会显示这一状态	无法在计算机上使用该驱动器, 应该尝试使用其他驱动器

10.2.3 修改驱动器分区风格

在计算机上安装了驱动器后, 使用时需要对其进行配置, 主要是对其进行分区并在分区上创建文件系统。磁盘使用的分区类型有主引导记录 (MBR) 与GUID分区表 (GPT) 两种。

1. MBR与GPT分区风格

MBR包含了一个分区表, 其中描述了分区在磁盘中的位置。如果磁盘采用的是这种分区风格, 则磁盘上的第一个扇区包含了主引导记录与一个二进制代码文件 (称为主引导代码), 用于引导系统。出于保护系统的需要, 该扇区不参加分区, 并且是不可见的。

采用MBR分区类型时, 磁盘支持最大4T的卷, 并使用如下两种分区类型的一种:

- 主分区;
- 扩展分区。

MBR驱动器最多可以包含4个主分区, 或者3个主分区与1个扩展分区。主分区是可以直接访问的驱动器部分 (用于文件存储等), 通过在其上创建文件系统, 用户就可以访问主分区。与主分区不同的是, 扩展分区不能直接访问, 但可以使用一个或多个逻辑驱动器 (用于存储文件) 对其进行配置。由于可以将扩展分区分配给多个逻辑驱动器, 因此可以将物理驱动器分配为多于4个部分。

32位与64位的Windows Server 2008与Windows Vista都支持MBR与GPT。GPT最初是为基于安腾处理器的高性能计算机设计的, 对大于2TB的磁盘 (基于X86与X64系统) 或基于安腾处理器的计算机,

建议使用GPT。MBR与GPT的关键区别在于分区数据的存储方式。在GPT风格中，关键性的分区数据存储在单独的分区中，并且为了增强的结构完整性，还使用了冗余的主分区与备份分区表。

基于GPT的磁盘有两个必不可少的分区，一个数据分区，一个或多个可选的（OEM或数据）分区：

- EFI系统分区（ESP）；
- 微软保留分区（MSR）；
- 至少一个数据分区。

此外，GPT磁盘支持最大18EB的卷、128个分区。需要指出的是，尽管GPT与MBR这两种磁盘风格有底层的区别，但大多数磁盘相关的任务是以相同方式执行的。

2. 转换分区表风格

通过使用CONVERT命令，DiskPart可以实现分区表风格在MBR与GPT之间的转换。在完成如下任务时，改变分区表风格是有用的。

- 将磁盘在不同计算机上使用，需要使用不同的分区表风格。
- 磁盘使用错误的分区表风格进行了格式化，需要进行转换。

要记住的是，只能在空磁盘上进行分区表风格的转换。这意味着磁盘或者是新的，或者是新格式化的。当然，你也可以通过移除待转换磁盘上的所有现存分区或卷，以便清空磁盘。

注解 DiskPart提供了一条CLEAN命令，可用于擦除磁盘上所有卷或分区信息。对焦点所在磁盘使用CLEAN命令后，其上所有分区或卷信息将会被清空。对MBR磁盘，这意味着MBR分区与隐藏扇区信息被重写；对GPT磁盘，这意味着GPT分区信息，包括受保护的MBR，都被重写。你也可以使用CLEAN ALL命令，该命令将磁盘上每个扇区填充为0。

警告 在将待转换驱动器上的数据进行备份之前，不要删除任何分区或卷，那会清空磁盘上所有数据。

要转换分区表风格，可以采用如下步骤。

- (1) 在命令提示符中，键入**diskpart**来调用DiskPart解释器。
- (2) 选择待操作的磁盘，将焦点放置在该磁盘上，比如：
DISKPART> select disk 2
- (3) 对磁盘进行转换，如下所示。
 - 要将磁盘从MBR转换为GPT，在命令提示符中键入**convert gpt**命令。
 - 要将磁盘从GPT转换为MBR，在命令提示符中键入**convert mbr**命令。

10.3 操作基本磁盘与动态磁盘

Windows Server 2008与Windows Vista支持下面两种类型的硬盘配置。

- **基本的。**以前Windows版本支持的标准磁盘类型，进行了分区后，基本磁盘可以在以前与当前的Windows版本中使用。
- **动态的。**一种增强的磁盘类型，可以在不需重启的情况下进行更新（大多数情况下）。动态磁盘可以分配为一个或多个卷，也可以使用软RAID进行配置。

注解 不能在便携式计算机或可移动介质上使用动态磁盘。

10.3.1 理解基本磁盘与动态磁盘

从以前的Windows版本升级到Windows Server 2008与Windows Vista时，磁盘会被初始化为基本磁盘；在新系统与未分区驱动器上安装Windows Server 2008与Windows Vista时，可以选择将驱动器初始化为基本的或动态的。

要注意的是，不能使用基本磁盘来创建容错的驱动器集。有鉴于此，如果想构建软RAID，就必须转换为动态磁盘，之后创建使用镜像或带区卷。容错以及无需重启计算机来修改磁盘是动态磁盘区别于基本磁盘的关键功能。

尽管可以在同一台计算机上同时使用基本磁盘与动态磁盘，但可以执行的磁盘配置任务是不同的。对于基本磁盘，可以操作分区，并执行如下一些任务。

- 格式化分区，并将其标记为活动分区。
- 创建或删除主分区与扩展分区。
- 创建或删除扩展分区内的逻辑驱动器。
- 将基本磁盘转换为动态磁盘。

对于动态磁盘，可以对卷进行操作，并执行如下一些任务。

- 创建标准卷与容错卷。
- 从镜像卷中移除一个镜像。
- 扩展简单卷或扩展卷。
- 将一个卷分割为两个卷。
- 修复镜像卷或RAID-5卷。
- 重新激活状态为丢失或脱机的磁盘。
- 从动态磁盘反转换为基本磁盘（进行此操作之前需要删除所有现存卷）。

对这两种磁盘类型，都可以执行如下一些任务。

- 查看磁盘、分区或卷的属性。
- 分配驱动器盘符。
- 配置安全性与共享驱动器。

不管是基本磁盘还是动态磁盘，都要记住下面5种特殊类型的驱动器部分。

- **活动分区**。活动分区或卷是用于系统缓存与启动的驱动器部分，有些带有可移动存储介质的设备也可能被标记为包含活动分区。
- **引导分区**。引导分区或卷包含了操作系统及其支持文件，系统分区（或卷）与引导分区（或卷）可以是相同的。
- **崩溃转储**。在系统崩溃事件发生时，计算机会向本分区写入转储文件。默认情况下，转储文件写入到%SystemRoot%文件夹，但也可以写入到任意需要的分区或卷上。
- **页面文件**。包含了操作系统使用的页面文件的分区。由于计算机可以将内存分页到多个磁盘，根据虚拟内存的配置方式，计算机可以包含多个页面文件分区或卷。
- **系统分区**。系统分区或卷包含了硬件特定的文件（用于加载操作系统），系统分区或卷不可以作为带区卷或跨区卷的一部分。

10.3.2 设置活动分区

在分区风格为MBR的磁盘上，可以将分区标记为活动分区，这意味着计算机是从该部分启动的，但不能将动态磁盘或卷标记为活动分区。在将包含活动分区的基本磁盘转换为动态磁盘时，该分区会转换为一个简单卷，并自动变为活动分区。在将分区标记为活动分区之前，要确保主分区（试图将该分区标记为活动分区）上包含了必要的启动文件。要指定活动分区，可以遵循如下几个步骤。

(1) 在命令提示符中，键入**diskpart**来调用DiskPart解释器。

(2) 选择包含了待标记为活动分区的磁盘，比如：

```
DISKPART> select disk 0
```

(3) 在命令提示符中键入命令**list partition**，列出磁盘上的分区。

(4) 选择待操作的分区，比如：

```
DISKPART> select partition 0
```

(5) 在命令提示符中键入命令**active**，使得选定的分区变为活动分区

警告 上面步骤的磁盘编号、分区编号是随意指定的，只是为了展示这一过程。要确保步骤2、4中对磁盘与分区的选择是正确的。如果错误地将一个不包含操作系统启动文件的分区标记为活动分区，则可能导致计算机无法启动。

10.3.3 改变磁盘类型：基本磁盘与动态磁盘的互相转换

Windows Vista与Windows Server 2008都同时支持基本磁盘与动态磁盘两种磁盘类型。有时候，可能需要将磁盘类型在二者之间进行转换，Windows提供了相应的工具来完成这一任务。在将基本磁盘转换为动态磁盘时，分区会自动转换为适当类型的卷。但无法再将其转换为基本磁盘上的分区，而必须在动态磁盘上删除这些卷，之后将磁盘类型转换为基本磁盘。删除卷会损毁磁盘上存储的所有信息。

1. 转换基本磁盘

将基本磁盘转换为动态磁盘是直接的，但包含很多约定。在开始之前，了解一下如下一些知识。

- 对MBR磁盘，要确保磁盘末尾有1MB的空闲空间。如果没有这样的空闲空间，会导致磁盘转换失败。磁盘管理工具与DiskPart都会自动保留这一空闲空间，在使用第三方磁盘管理工具时，则需要关注是否保留了这一空闲空间。
- 对GPT磁盘，必须包含临近的、可识别的分区。如果GPT磁盘包含了Windows不能识别的分区，比如由其他操作系统创建的分区，则无法将其转换为动态磁盘。

此外，两种类型的磁盘都遵循如下一些准则。

- 不能对扇区大小超过512字节的驱动器进行转换。对这种驱动器，在进行转换之前需要重新格式化。
- 在便携式计算机或移动设备中不能使用动态磁盘。对这种驱动器，只能将其标记为基本驱动器（带有主分区）。
- 如果系统或引导分区是跨区卷、带区卷、镜像或RAID-5卷的一部分，则不能进行磁盘转换。转换之前，应该终止跨区卷、带区卷、镜像。
- 对于其他类型的分区，即便是跨区卷、带区卷、镜像或RAID-5卷的一部分，也可以进行转换。但必须对驱动器集中的所有驱动器一起转换，转换之后，变为动态卷。

通过如下步骤，可以将基本磁盘转换为动态磁盘。

- (1) 在命令提示符中，键入diskpart来调用DiskPart解释器。
- (2) 选择要转换为动态磁盘的磁盘，比如：

DISKPART> select disk 0

- (3) 在命令提示符中，键入convert dynamic进行磁盘转换。

2. 转换动态磁盘

将基本磁盘转换为动态磁盘后，再转换为基本磁盘的唯一途径是移除磁盘上的所有卷。这将确保磁盘被清空，其上所有数据被移除。DiskPart提供了一条名为CLEAN的命令，用于擦除磁盘上所有卷或分区信息。选定了一个磁盘并将焦点放置到其上之后，键入CLEAN命令，则该磁盘上所有分区或卷信息都将被清除。

在MBR磁盘上，这意味着MBR分区与隐藏扇区信息被重写；对GPT磁盘，这意味着GPT分区信息，包括受保护的MBR，都被重写。也可以使用CLEAN ALL命令，该命令将磁盘上所有扇区填充为0，从而完全删除磁盘上所有数据。

通过如下步骤，可以将空动态磁盘转换为基本磁盘。

- (1) 在命令提示符中，键入diskpart来调用DiskPart解释器。
- (2) 选择要转换为基本磁盘的磁盘，比如：

DISKPART> select disk 0

- (3) 在命令提示符中，键入convert basic进行磁盘转换。

完成上述步骤之后，动态磁盘就转换为基本磁盘，之后就可以在该磁盘上创建新分区与逻辑卷。

10.4 磁盘维护

很多命令行工具可以用于磁盘维护，包括FSUtil、Chkdsk以及Defrag等。

10.4.1 使用 FSUtil 获取磁盘信息并管理文件系统

文件系统工具（FSUtil）是本书尚未讲解的一个工具。

1. FSUtil概览

FSUtil的命令结构相当复杂，但你需要做的仍然是键入命令字符串。其中包含了命令以及相关的子命令，以便让FSUtil去完成相关任务。表10-3中总结了可用的FSUtil命令。

表10-3 FSUtil命令及其使用方法

BEHAVIOR	使用相关的子命令查看并控制短文件名（MS-DOS）的生成方式，NTFS卷上最后一次访问的时间戳是否已更新，配额事件写入到系统日志的频率，NTFS换页池内存与NTFS非换页池内存的内部缓存级别，以及为Master文件表（MFT）保留的磁盘空间总量
DIRTY	使用相关的子命令查询或设置卷的dirty位。如果某个卷是dirty，则说明该卷可能出错，下一次计算机启动时会运行一个名为AUTOCHECK的程序对磁盘进行检查。如果必要，还会运行磁盘检查来修复存在的错误
FILE	使用相关的子命令根据用户名（只适用于磁盘配额激活的情况）寻找文件，检查文件的稀疏区域，设置文件的有效数据长度，取消文件的稀疏部分
FSINFO	使用相关子命令列出计算机上的驱动器、查询驱动器类型、获取卷信息

(续)

HARDLINK	使用相关子命令创建硬链接,以便单一文件可以在多个目录中出现(甚或在同一个目录中以多个文件名的形式出现)。程序可以打开任意的链接来修改文件,只有在所有链接都删除的情况下,该文件才会从文件系统中删除
OBJECTID	使用相关子命令管理文件与目录的对象标识符
QUOTA	使用相关子命令管理NTFS卷上的磁盘配额
REPARSEPOINT	使用相关子命令查看或删除稀疏点,稀疏点主要用作目录连接点与卷挂接点
REPAIR	使用相关子命令查看并修改自愈状态,默认情况下,自愈在Windows Server 2008与Windows Vista中都是激活的
RESOURCE	使用相关子命令管理事务资源管理者。事务是被作为单一业务单元处理的一系列操作,这些操作或者都发生,或者都不发生
SPARSE	使用相关子命令管理稀疏文件,稀疏文件是指其中包含了一个或多个未分配数据区域的文件
TRANSACTION	使用相关子命令管理事务
USN	使用相关子命令管理更新序列号(USN)修改日志,USN修改日志提供了对卷上文件所有修改操作进行记录的持久化日志
VOLUME	使用相关子命令卸载一个卷,或者查询还有多少可用的空闲空间。

2. 使用FSUtil

尽管FSUtil有很多高级应用,比如移除磁盘上的稀疏点、管理磁盘配额、指定稀疏文件等,但也有很多基本的应用,这些基本应用对获取磁盘信息是有益的。

获取某计算机上的驱动器列表。要根据盘符获取计算机上的驱动器列表,可以键入如下命令:

fsutil fsinfo drives

这一命令的输出信息以字母顺序展示了计算机上的可用驱动器,如下:

Drives: A:\ C:\ D:\ F:\ G:\ T:\ U:\

获取驱动器类型。获知驱动器之后,可以键入命令**fsutil fsinfo drivetype**(其后跟随具体的驱动器),以便获取某个具体驱动器的类型信息,比如:

```
C:\>fsutil fsinfo drivetype g:
g: - CD-ROM Drive
```

输出信息表明,G:盘是一个CD-ROM驱动器。当然,你也可以使用DiskPart的list volume命令,获取计算机上所有磁盘的类似信息,但FSUtil提供了另一种有益的途径。

获取驱动器的详细信息。要获取某个驱动器的详细信息,可以键入**fsutil fsinfo volumeinfo**(其后跟随具体的驱动器),比如:

```
C:\>fsutil fsinfo volumeinfo c:
```

执行后,FSUtil会列出卷名、序列号、文件系统类型以及支持的功能,比如:

```
Volume Name : Primary
Volume Serial Number : 0x23b36g45
Max Component Length : 255
File System Name : NTFS
Supports Case-sensitive filenames
Preserves Case of filenames
```

Supports Unicode in filenames
 Preserves & Enforces ACL's
 Supports file-based Compression
 Supports Disk Quotas
 Supports Sparse files
 Supports Reparse Points
 Supports Object Identifiers
 Supports Encrypted File System
 Supports Named Streams
 Supports Transactions

获取驱动器的扇区与簇信息。要获取某个NTFS磁盘的扇区或簇信息，可以键入**fsutil fsinfo ntfsinfo**（其后跟随具体的驱动器），比如：

```
C:\>fsutil fsinfo ntfsinfo c:
```

执行后，FSUtil会列出扇区数与簇数，包括总簇数、空闲簇数、保留簇数等信息，比如：

```

NTFS Volume Serial Number :          0x23b36g45
Version :      3.1
Number Sectors :          0x0000000008fcf7c3
Total Clusters :          0x0000000000eb9f38
Free Clusters :          0x0000000000d12400
Total Reserved :          0x0000000000000000
Bytes Per Sector :          512
Bytes Per Cluster :          4096
Bytes Per FileRecord Segment :          1024
Clusters Per FileRecord Segment :          0
  
```

获取驱动器的空闲空间信息。要获取某磁盘上可用空间总量，可以键入**fsutil volume diskfree**（其后跟随具体的驱动器），比如：

```
C:\>fsutil volume diskfree c:
```

执行后，FSUtil会列出磁盘的总字节数、空闲字节数、可用的空闲字节数等信息，比如：

```

Total # of free bytes          : 52231667712
Total # of bytes              : 60028059648
Total # of avail free bytes    : 52231667712
  
```

确定某个卷是否dirty。要确定某磁盘是否已被标记为**dirty**，可以键入**fsutil dirty query**（其后跟随具体的驱动器），比如：

```
fsutil dirty query c:
```

如果该卷上存在需要修复的错误（或已被标记为**dirty**），则FSUtil会报告如下信息：

```
Volume - c: is Dirty
```

如果该卷上不存在任何已知的错误，则FSUtil会报告如下信息：

```
Volume - c: is NOT Dirty
```

10.4.2 检查磁盘的错误与坏扇区

如果需要检查磁盘错误与坏扇区，可以使用命令行工具Check Disk（Chkdsk.exe）。它可以检查基本磁盘与动态磁盘的完整性，也可以用于检查与修复（可选）FAT、FAT32、NTFS卷上发现的问题。

Chkdsk.exe可以检查与纠正很多类型的磁盘错误。它主要用来寻找文件系统及其相关元数据的不一致性，且其用于定位错误的一种途径是，将卷位图与文件系统中分配给文件的磁盘扇区进行比较。然而，chkdsk.exe不能修复文件中看起来结构上比较完整的损坏数据。

分析磁盘但不修复它

在命令行中键入命令名，其后跟随驱动器盘符与一个分号，就可以测试驱动器的完整性。比如，要检查C盘驱动器的完整性，可以键入命令**chkdskc:**。对于FAT与FAT32卷，Check Disk会报告类似于如下的信息：

```
The type of the file system is FAT32.
The volume is in use by another process. Chkdsk
might report errors when no corruption is present.
Volume TRAVELDRIVE created 4/21/2008 12:32 AM
Volume Serial Number is DFGA-9871
Windows is verifying files and folders...
File and folder verification is complete.
Windows has checked the file system and found no problems.
    8,043,504 KB total disk space.
        20 KB in 5 folders.
    3,739,684 KB in 85 files.
    4,303,796 KB are available.

    4,096 bytes in each allocation unit.
    2,010,876 total allocation units on disk.
    1,075,949 allocation units available on disk.
```

上面的命令对文件分配表中的每一条记录进行一致性检查，对当前分配的文件与文件夹记录进行分析，并使用根目录表确定每个记录的起始簇。该工具检查每个文件，并注意查看输出信息中的差异。那些标记为已被文件或文件夹使用但实际上没有实际应用的簇都会被显示出来，在修复过程中，这些簇可以标记为可用。输出信息中标记的任意其他的差异也可以在修复过程中修订。

对于NTFS，chkdsk.exe在多个阶段进行分析，并分阶段报告其分析进程。下面给出了一个对NTFS卷进行检查时的输出示例：

```
The type of the file system is NTFS.
WARNING! F parameter not specified.
Running CHKDSK in read-only mode.

CHKDSK is verifying files (stage 1 of 3)...
    73088 file records processed.
File verification completed.
    123 large file records processed.
    0 bad file records processed.
    0 EA records processed.
    0 reparse records processed.
CHKDSK is verifying indexes (stage 2 of 3)...
    269869 index entries processed.
Index verification completed.
    5 unindexed files processed.
CHKDSK is verifying security descriptors (stage 3 of 3)...
    73088 security descriptors processed.
Security descriptor verification completed.
    3511 data files processed.
```

```
CHKDSK is verifying Usn Journal...
26911912 USN bytes processed.
Usn Journal verification completed.
Windows has checked the file system and found no problems.
```

```
488384000 KB total disk space.
353648812 KB in 69111 files.
25836 KB in 3512 indexes.
0 KB in bad sectors.
180660 KB in use by the system.
65536 KB occupied by the log file.
134528692 KB available on disk.
4096 bytes in each allocation unit.
122096000 total allocation units on disk.
33632173 allocation units available on disk.
```

可以看出, `chkdsk.exe`操作是分3个阶段进行的。在第一个阶段, `chkdsk.exe`的作用是对文件结构进行验证:

```
CHKDSK is verifying files (stage 1 of 3)...
73088 file records processed.
File verification completed.
123 large file records processed.
0 bad file records processed.
0 EA records processed.
0 reparse records processed.
```

这意味着`chkdsk.exe`对MFT中的每一个文件的记录进行一致性检查, 对当前已分配的所有文件记录进行分析, 确定文件记录存储在哪些簇中, 并将其与卷的簇位图元数据文件进行比较, 任意的差异都会在其输出信息中进行标记。比如, 那些标记为已被文件或文件夹使用但实际上没有应用的簇都会被显示出来, 在修复过程中, 这些簇可以标记为可用。

在第二阶段中, `chkdsk.exe`对磁盘索引条目进行确认:

```
CHKDSK is verifying indexes (stage 2 of 3)...
269869 index entries processed.
Index verification completed.
5 unindexed files processed.
```

这意味着`chkdsk.exe`通过检查目录索引(从卷的根目录索引开始, 且存储于索引元数据文件中)来验证目录结构。通过对目录索引的检查, `chkdsk.exe`可以确认是否每个索引记录都对应了磁盘上真实存在的目录, 是否每个被认为存在于某个目录中的文件确实存在于某个目录中。此外, `chkdsk.exe`还对包含MFT目录但实际上并不存在于该目录中的文件进行检查, 在修复过程中, 这些丢失的文件可以恢复。

如果在验证磁盘索引条目的过程中发现丢失的文件, 并且提供了/F参数, 则`chkdsk.exe`会尝试对这些文件进行恢复。典型情况下, 恢复的文件在相关的磁盘驱动器的根目录下保存为.chk文件。如果`chkdsk.exe`发现了未索引的文件, 则会对其进行索引编目。

在第三阶段, `chkdsk.exe`会对安全描述符进行验证:

```
CHKDSK is verifying security descriptors (stage 3 of 3)...
73088 security descriptors processed.
Security descriptor verification completed.
```

3511 data files processed.

这意味着chkdsk.exe对卷上每个文件与目录对象的安全描述符进行验证,这是通过安全元数据文件检测安全描述符是否正常工作实现的,但实际上并不真正检查安全描述符中标记的用户或组是否存在。

chkdsk.exe还会对USN 修改日志中的更新序列号进行验证。USN修改日志对卷中所有相关操作进行了完整的记录,记录了所有增加、删除与修改等操作,而不管是哪个用户以哪种方式进行的这些操作。这一日志是持久性的,并不会随着关机或重启而消失。如果在对USN 修改日志进行验证时发现USN错误,则chkdsk.exe会尝试对其进行修复。

完成上述3个步骤后,chkdsk.exe会报告是否存在问题,并提供一份关于磁盘的报告信息:

Windows has checked the file system and found no problems.

```
488384000 KB total disk space.
353648812 KB in 69111 files.
 25836 KB in 3512 indexes.
   0 KB in bad sectors.
180660 KB in use by the system.
 65536 KB occupied by the log file.
134528692 KB available on disk.

4096 bytes in each allocation unit.
122096000 total allocation units on disk.
33632173 allocation units available on disk.
```

10.4.3 修正磁盘错误

对磁盘进行分析时,默认情况下并不会修复存在的错误。如果希望在检测的过程中自动地对其中存在的错误进行恢复,就需要使用/F参数:

```
chkdsk /f C:
```

要注意的是,chkdsk.exe不能对使用中的卷进行修复。如果卷正处于使用中,则chkdsk.exe会弹出提示信息,询问是否希望在下次重启系统时对其进行修复。此外,使用/R参数或/X参数可以替代/F参数的功能(/X参数只适用于NTFS卷)。**/R**参数用于定位磁盘的坏扇区,并恢复可读的信息,**/X**参数用于强制卸载NTFS卷(如果需要)。

真实场景 如果指定了/R参数,则chkdsk.exe在分析与修复的过程中会增加一个附加的步骤,即对磁盘上的每一个扇区进行检查,以便确定这些扇区是否可以正确进行读写操作。如果某扇区是正在使用中的簇的一部分,则chkdsk.exe会将该簇中的正常数据移动到新簇中。

坏扇区中数据可以恢复的前提是,存在冗余的数据副本,并可以复制出来。检测之后,坏扇区将不再用于存储数据,因此至少可以保证将来不会引发问题。对磁盘上每个扇区都进行检查是一个耗时的过程,有鉴于此,典型情况下都是使用Chkdsk /F参数来检测与修复常见的错误,而不使用Chkdsk /R。

对NTFS卷,可以强制chkdsk.exe重新评估那些使用/B参数标记为bad的簇,**/B**参数暗含了/R参数的功能。重新评估时,chkdsk.exe会再一次确定标记为bad的簇是否可以正常读写。如果可以,则将该簇标记为good,以便磁盘子系统可以使用该簇。

通过使用/V参数,可以使chkdsk.exe显示更多的信息。对NTFS卷,通过使用/I参数,可以使

chkdsk.exe只对磁盘索引条目进行有限的检查；通过使用/C参数，可以使chkdsk.exe跳过对文件夹结构中循环错误的检查。所谓循环错误，实际上是一种极少发生的错误类型，发生这种错误时，目录中会包含一个指向自身的指针，导致无限循环。

要了解如何使用chkdsk.exe，可以参考如下一些实例。

发现并修复C盘驱动器上的错误：

```
chkdsk /f C:
```

定位并修复C盘驱动器上的坏扇区：

```
chkdsk /r C:
```

对C盘驱动器（NTFS卷）进行最小限度的检查：

```
chkdsk /i /c C:
```

10.4.4 对系统启动时的自动检测进行控制

默认情况下，Windows Server 2008与Windows Vista在启动时会对所有磁盘进行检查，并在必要的时候启动chkdsk.exe来修复存在的错误。有两个工具可用于对启动时的自动磁盘检查进行控制，分别是AUTOCHK与CHKNTFS。操作系统启动时，会启动自动检测，以便对驱动器进行自动检查。不能直接调用自动检测，但可以使用NTFS检测来控制其工作方式。通过NTFS检测，可以决定计算机下一次启动时是否对磁盘进行检查，也可以更改磁盘自动检查的选项。

1. 确定自动检测状态

要确定计算机下次启动时是否对某磁盘进行检查，可以键入：

chkntfs Volume:

其中，*Volume*是待检查的盘符，其后跟随一个分号，比如：

```
chkntfs c:
```

你也可以同时指定多个盘符，其间使用空格分隔开。下面的命令可以确定驱动器C、D、E的磁盘检测状态：

```
chkntfs c: d: e:
```

NTFS检测可以报告文件系统类型，以及磁盘状态为dirty还是not dirty，如下所示：

```
The type of the file system is NTFS.  
C: is not dirty.  
The type of the file system is NTFS.  
D: is dirty.
```

从上面信息可以看出，C盘驱动器的状态为not dirty，因而自动检测不会在其上触发磁盘检测。而D盘驱动器的状态为dirty，这表明该驱动器上可能存在错误，因而计算机启动时，自动检测会在其上触发磁盘检测。

2. 配置自动检测参数

通过使用NTFS检测，可以对自动检测的工作方式进行配置。计算机重新引导时，操作系统会显示一个倒计时定时器，使得用户可以在自动检测开始之前对其进行取消。通过/T参数，可以设置倒计

时定时器的时间长度，如下所示：

```
chkntfs /t:NumSeconds
```

其中，*NumSeconds*是该定时器设定的倒计时秒数，比如：

```
chkntfs /t:15
```

如果希望计算机启动时不对某个或某几个卷进行检查（即便该卷标记为需要检查），可以使用/X参数，其后跟随盘符，比如：

```
chkntfs /x d: e:
```

这一命令使得计算机不会对D盘与E盘进行检查，即便这两个磁盘标记为dirty。

如果希望计算机启动时对某个或某几个卷进行检查（这也是自动检测的默认配置），可以使用/C参数，其后跟随盘符，比如：

```
chkntfs /c c: d:
```

这一命令使得计算机启动时对D盘与C盘进行检查，用于确定这两个磁盘标记为dirty还是not dirty。

最后一个可用的参数是/D，用于恢复自动检测的默认配置（倒计时定时器除外），即计算机启动时对所有磁盘驱动器进行自动检查。

10.5 磁盘碎片整理

进行文件添加、移动或删除时，计算机磁盘驱动器上的数据会变得碎片化。磁盘上碎片数据较多时，大文件无法写入到磁盘上单一的连续区域。由此导致的结果是，大文件被分散写到磁盘上很多小区域中，从而导致文件读取时花费更多的时间开销。要减少磁盘中的碎片，应该定期对其进行分析与碎片整理。尽管可以将Windows Server 2008与Windows Vista配置为自动进行磁盘碎片整理，但也可以使用Defrag这一命令行工具来检测FAT、FAT32、NTFS卷上的碎片情况，并根据需要对其进行碎片整理。

1. 理解与使用Defrag

典型情况下，碎片整理的过程分为两个步骤。首先，分析磁盘的碎片化程度，确定其是否需要行碎片整理，之后根据分析结果决定下一步。实际使用时，可以键入defrag命令（其后跟随盘符与分号）来完成这两个步骤，比如：

```
defrag c:
```

Defrag会对磁盘进行分析，如果分析结果表明该磁盘需要进行碎片整理，则开始这一过程；如果不需要，则Defrag在分析之后不进行整理，并报告说磁盘不需要进行碎片整理。

如果需要对磁盘进行彻底的碎片整理，一般要求磁盘上至少有15%的空闲空间。Defrag会使用这一部分空间作为文件碎片的存储区域。如果磁盘空闲空间小于15%，则只能进行部分的碎片整理。此外，不能对已标记为dirty的磁盘进行碎片整理，因为这意味着磁盘上存在错误。对这种情况，必须先运行磁盘检测工具来修复其上的错误。

Defrag有几个可用的参数，比如-a，作用是只对磁盘进行分析，但不对其进行碎片整理。Defrag有两种碎片整理模式：部分整理，由-r标记指定；全面整理，由-w标记指定。默认情况下，在文件碎片小于64MB时，Defrag会进行部分的碎片整理。使用-w标记后，Defrag会进行全面的碎片整理，而不管文件碎片的大小。

Defrag还有两个参数：-v，用于指定verbose输出；-f，强制进行碎片整理（即便自由空间较小），

但强制进行碎片整理可能需要花费很长的时间，并且也不能保证碎片整理的完全性。

2. 只进行Defrag分析

有时候，只需要对磁盘进行分析，以便确定以后是否对其进行碎片整理。Defrag有两种分析模式：概要模式，由-a标记指定，且不使用-v标记；全模式，使用-a标记，同时使用-v标记。

要对磁盘进行概要分析，而不对其进行碎片整理，可以键入**defrag -a**命令，其后跟随盘符与分号。分析之后，Defrag会报告磁盘是否应该进行整理。下面实例中，分析之后，报告说该磁盘不需要进行碎片整理：

```
C:\>defrag -a c:
```

Windows磁盘碎片整理程序

版权所有 (c) 2006 Microsoft Corp.

卷C分析报告

卷大小 = 457 GB

可用空间 = 358 GB

最大可用空间扩展 = 209 GB

文件碎片百分比 = 2 %

注意：在NFS卷上，大小超过64MB的文件碎片不包含在碎片统计信息中。

不需要对该卷进行碎片整理。

可以看出，该磁盘只有2%的碎片率，因而不需要进行碎片整理。下面的实例则展示了一个碎片化程度很高的磁盘：

```
C:\>defrag -a d:
```

Windows磁盘碎片整理程序

版权所有 (c) 2006 Microsoft Corp.

卷D分析报告

卷大小 = 446 GB

可用空间 = 131 GB

最大可用空间扩展 = 74.69 GB

文件碎片百分比 = 29 %

注意：在NFS卷上，大小超过64MB的文件碎片不包含在碎片统计信息中。

你应该对该卷进行碎片整理。

分析之后，Defrag建议对磁盘进行碎片整理，可以根据系统维护需要进行。

要对磁盘进行全面分析，但不对其进行碎片整理，可以键入**defrag -a -v**命令，其后跟随盘符与分号。分析之后，Defrag会报告磁盘是否应该进行整理。下面实例中，分析之后，报告说该磁盘需要进行碎片整理：

```
C:\>defrag -a -v d:
```

Windows磁盘碎片整理程序

版权所有 (c) 2006 Microsoft Corp.

卷D分析报告

卷大小 = 466 GB

簇大小 = 4 KB

已使用空间 = 355 GB

可用空间 = 131 GB

可用空间百分比	= 28 %
文件碎片	
文件碎片百分比	= 29 %
可移动文件总数	= 72,608
文件平均大小	= 5 MB
碎片文件总数	= 18,200
剩余碎片总数	= 91,257
每个文件的碎片平均个数	= 1.64
不可移动文件总数	= 10
可用空间碎片	
可用空间	= 131 GB
可用空间扩展总数	= 552
每个扩展的可用空间平均大小	= 243 MB
最大可用空间	= 74.69 GB
文件夹碎片	
文件夹总数	= 3,504
零碎文件夹	= 15
剩余文件夹碎片	= 35
宿主文件表 (MFT) 碎片	
MFT总大小	= 71 MB
MFT记录计数	= 72,757
使用中的MFT百分比	= 99
总的MFT碎片	= 8

注意：在NFS卷上，大小超过64MB的文件碎片不包含在碎片统计信息中。

你应该对该卷进行碎片整理。

当前分析卷的类型决定了报告中的不同区域，报告区域包含如下一些要素。

- **文件碎片**。提供了对文件级碎片的整体描述，包括使用中磁盘空间的碎片率，卷上可移动文件总量，这些文件的平均大小，碎片化文件总量，更多的碎片总量，每个文件的平均碎片数，不可移动文件总量等。理想情况下，整体碎片化程度应该小于等于10%，而每个文件的碎片化程度接近100%。
- **空闲空间碎片**。提供了对卷上未使用磁盘空间碎片的整体描述，包括卷上有多少可用的空闲空间，空闲空间定位所在的范围总数，每个范围上的平均空闲空间量，以及最大空闲空间范围。
- **文件夹碎片**。提供了对文件夹级碎片的整体描述，包括卷上文件夹总量，存在碎片的文件夹总量等。
- **主文件表 (MFT) 碎片**。只适用于NFS卷，提供了对MFT碎片的整体描述，包括MFT当前大小，其中包含的记录数，使用中MFT的百分比，MFT中碎片总量等。在上面的实例中，MFT中存在一些碎片。但应该引起注意的是MFT使用率已达到其最大值的99%，这会导致MFT碎片化程度随时间不断增加。鉴于该卷上还有28%的空闲空间，MFT碎片将有可能分布到这一部分。

鉴于上面的分析结果，应该再次运行Defrag（而不需要指定-a参数），并且使用-w参数进行全面的碎片整理（而不是默认情况的部分碎片整理）。尽管每一种技术都不能消除所有碎片，但都会有所帮助，使得磁盘空间的使用更加高效。对存在碎片的卷，进行碎片整理后，会发现性能有所提升。

安装新计算机或对现有计算机进行升级时，通常需要对计算机上的硬盘驱动器进行分区。

DiskPart可以使用主引导记录（MBR）或GUID分区表（GPT）这两种分区风格。使用MBR时，磁盘驱动器最多可以包含4个主分区，或3个主分区与1个扩展分区。在Windows Vista与Windows Server 2008上使用GPT时，可引导磁盘或系统磁盘上必须包含两个分区，即EFI系统分区与微软保留分区。此外，还可以包含1个或多个可选的OEM分区与数据分区，分区总量最多可达128个。

11.1 获取分区信息

使用DiskPart时，可以使用LIST PARTITION命令获取选定磁盘的分区信息。如下面实例所示，LIST PARTITION列出了选定磁盘上所有分区的信息。比如，如果键入**select disk 2**，之后键入**list partition**，就可以看到disk 2上分区列表：

Partition ###	Type	Size	Offset
Partition 1	Primary	706 MB	32 KB
Partition 2	Primary	706 MB	706 MB
Partition 3	Primary	706 MB	1412 MB
Partition 4	Extended	1004 MB	2118 MB
Partition 5	Logical	502 MB	2118 MB
Partition 6	Logical	502 MB	2620 MB

注解 输出信息中相应条目前的星号表示该分区是当前选定的分区，焦点放置在其上。

从输出信息中可以看出，LIST PARTITION命令会展示如下一些要素。

- **Partition ###**。分区编号，可以使用**select partition n**来选中并操作某个分区。
- **Type**。布局类型，分区布局包括主要分区、扩展分区与逻辑分区。
- **Size**。分区的总容量。
- **Offset**。该分区的字节偏移量，总是在最近的柱面边界附近。

注解 柱面是分区中某个磁盘驱动器的一部分，柱面进一步划分为磁道，磁道进一步划分为扇区，扇区划分为字节，字节是磁盘中进行数据存储的最小单位。比如，一个4GB大小的磁盘可以包含525个柱面，每个柱面可以包含255个磁道，每个磁道可以包含63个扇区，每个扇区包含512个字节。本例中，柱面大小为8MB，因此，分区的字节偏移量总是在8MB上下。但磁盘上的第一个扇区是一个例外情况，该扇区以第一个可用柱面的头部开始。

11.2 创建分区

在基本磁盘上创建分区的方式取决于分区风格，鉴于在MBR磁盘与GPT磁盘创建分区的类型不一样，下面将对在MBR磁盘与GPT磁盘创建分区分别进行讲述。

11.2.1 在 MBR 磁盘上创建分区

对于MBR磁盘，可以使用DiskPart创建主分区与扩展分区。主分区可以占据整个磁盘，也可以根据需要设定合适的大小。每个物理驱动器可以包含一个扩展分区，扩展分区可以包含1个或多个逻辑驱动器，逻辑驱动器实际上是带有自身文件系统的分区部分。尽管你可以根据需要对逻辑驱动器大小进行设置，但也要考虑在当前工作站或服务器上如何使用逻辑驱动器。通常，使用逻辑驱动器的一个目标是将容量非常大的物理驱动器划分为便于管理的几个部分。根据这一出发点，可以将60GB的扩展分区划分为3个大小分别为20GB的逻辑驱动器。

1. 创建主分区

在向磁盘上添加主分区之前，应该对磁盘上的空闲空间总量进行评估，并检查当前的分区配置。要创建主分区，可以遵循如下步骤。

(1) 在命令提示符中键入**diskpart**，调用DiskPart。

(2) 键入命令**list disk**，列出计算机上的磁盘，并检查空闲空间：

Disk ###	Status	Size	Free	Dyn	Gpt
-----	-----	----	----	---	---
Disk 0	Online	56 GB	0 B		
Disk 1	Online	29 GB	0 B		
Disk 2	Online	37 GB	37 GB		

(3) 本例中，disk 2有37GB的空闲空间可用于分区，该磁盘上尚无分区，因为磁盘容量也是37GB。如果磁盘容量与空闲空间大小不一致，则说明有一部分磁盘空间已经分配给了某个分区。如果需要操作disk 2，可以键入**select disk 2**命令来选定该磁盘，并使用**list partition**命令查看其上的分区。

选定了某磁盘并将焦点放置到其上后，可以通过如下命令创建主分区：

```
create partition primary size=n
```

其中，*n*为以MB为计数单位的空间大小。如果没有指定空间大小，则该分区将占据磁盘上所有未分配空间。

注解 本例中，分区是在磁盘上第一块空闲空间的起始处创建的。DiskPart会通过将偏移量参数设置为合适的值来自动做到这一点。要记住的重要一点是，偏移量是根据最近的柱面边界计算的。该值会根据与最近的柱面边界协调的原则调整，这有可能会改变分区或逻辑驱动器的最终大小。

提示 对于RAID逻辑单元数（LUN）阵列，你可能需要将卷或分区的所有起始点与最近的对其边界对其，来提高性能。要做到这一点，可以使用Align参数，其语法格式为align=*n*。其中，*n*为从磁盘起始处到最近的对其边界的KB数，比如align=64。

创建分区之后，焦点将自动地放置到其上，表示该分区已被选定，但该分区尚无盘符或挂载点，必须使用ASSIGN命令进行分配。要完成最后的创建工作，还必须使用FORMAT命令对其进行格式化。要了解更多信息，参阅11.3与11.4等节。

2. 在扩展分区上创建逻辑驱动器

每个磁盘驱动器可以包含一个扩展分区。与创建主分区类似，在向磁盘上添加扩展分区之前，应该对磁盘上的空闲空间总量进行评估，并检查当前的分区配置。之后，可以在选定的磁盘（焦点放置到其上）上的未分配空间上创建扩展分区，使用如下命令：

```
create partition extended size=n
```

其中， n 为以MB为计数单位的空间大小。如果没有指定空间大小，则该分区将占据磁盘上所有未分配空间。

创建扩展分区之后，焦点将自动地放置到其上，表示该分区已被选定。与创建主分区不同的是，不要直接对扩展分区进行盘符指定或格式化，而是在其上创建逻辑驱动器，之后对这些逻辑驱动器进行盘符指定或格式化。

要在扩展分区内创建逻辑驱动器，可以使用如下命令：

```
create partition logical size=n
```

其中， n 为以MB为计数单位的空间大小。如果没有指定空间大小，则该分区将占据扩展分区上所有未分配空间。指定逻辑驱动器大小时，要记住分区内所有逻辑驱动器大小的总量要小于扩展分区容量。这也是为什么在第10章，在创建了4096MB的扩展分区后，又在该分区内分别创建了2048MB与2047MB逻辑驱动器的原因。

创建逻辑驱动器之后，焦点将自动地放置到其上，但该分区尚无盘符或挂载点，必须使用ASSIGN命令进行分配。要完成最后的创建工作，还必须使用FORMAT命令对其进行格式化。

11.2.2 在 GPT 磁盘上创建分区

在GPT磁盘上，可以创建如下类型的分区。

- EFI系统分区（ESP）。
- 微软保留分区（MSR）。
- 主分区。
- 逻辑磁盘管理（LDM）元数据分区。
- LDM数据分区。
- OEM或未知分区。

与MBR磁盘类似，在GPT磁盘上创建分区之前，应该对磁盘上空闲空间总量进行评估，并检查当前分区配置。选定了要操作的磁盘并将焦点放置到其上之后，就可以创建需要的分区。

1. 创建EFI系统分区

带有EFI系统分区的计算机必须包含GPT磁盘（其中包含ESP），这一分区与X86计算机上的系统卷类似，因为此分区内包含了启动操作系统所必需的文件。Windows Server 2008在安装时会创建ESP，使用FAT对其进行格式化，并指定其大小，最小值为100MB，或者为磁盘大小的1%（最大不超过1000MB）。

在用于加载操作系统的磁盘上，EFI系统分区应该是第1个分区，MSR应该是第2个分区。在非系

统磁盘上，由于不用于启动操作系统的GPT磁盘不包含EFI系统分区，因此MSR应该是第1个分区。

通常，在带有EFI的计算机上安装Windows Server 2008时，ESP是自动创建的。但也有些情况下，在服务器上安装另外的GPT磁盘后，需要手动创建ESP。比如，使用新磁盘作为系统引导设备，而不再使用原有的引导设备时。

使用如下命令可以创建EFI系统分区：

```
create partition efi size=n
```

其中， n 为以MB为计数单位的空间大小。创建EFI之后，焦点将自动地被放置到其上，表示该分区已被选定。刚创建的分区尚未指定盘符与挂载点，但对EFI分区与数据分区来讲这是需要指定的。要完成EFI的创建，还必须对其进行格式化。通常，EFI分区格式化为FAT格式。

DiskPart可以自动设置偏移量参数，以便标记EFI系统分区。大多数场景下，不应该手动设置偏移量参数。EFI的全局唯一标识符（GUID）如下：

```
c12a7328-f81f-11d2-ba4b-00a0c93ec93b
```

真实场景 如果由于某些原因，GUID没有设置或设置错误，则可以使用SETID命令来正确标记分区，并将其重建为MSR分区。鉴于这种做法存在导致计算机失败或无法启动的风险，只有经验丰富并且对GPT磁盘有深刻理解的管理员才可以尝试使用SETID命令。进一步地说，使用SETID应该是不得已而为之的一种方法，比如，无法使用CREATE PARTITION MSR命令创建MSR分区。此外，SETID无法用于动态磁盘，只能由OEM使用。最后，要记住DiskPart不会对GUID进行检测以确保其有效性。

2. 创建微软保留分区

带有EFI的每个GPT磁盘上都必须包含一个MSR分区。MSR分区包含了操作系统在指定磁盘操作时所需的额外空间。比如，在将基本磁盘转换为动态GPT磁盘时，Windows操作系统需要使用MSR分区的1MB空间来创建LDM元数据分区，以便转换时使用。

Windows操作系统会自动创建MSR分区，对引导磁盘，MSR分区是在安装操作系统时与ESP一起创建的。在磁盘由MBR转换为GPT，或者访问尚未包含MSR分区的GPT磁盘时，都会自动创建MSR分区。

如果GPT磁盘包含了ESP，并将其作为磁盘上的第1个分区，则MSR分区通常就是磁盘上第2个分区；如果GPT磁盘尚未包含ESP，则MSR分区通常就是磁盘上第1个分区；如果磁盘起始处包含了一个主分区，则MSR分区应该放置在磁盘尾部。

MSR分区的大小需要根据相关磁盘大小进行设置，对最大16GB的磁盘，MSR分区大小通常为32MB。对其他磁盘，MSR分区通常为128MB。

使用如下命令可以创建MSR分区：

```
create partition msr size=n
```

其中， n 为以MB为计数单位的空间大小。创建EFI之后，焦点将自动地放置到其上，表示该分区已被选定。刚创建的分区尚未指定盘符与挂载点。Windows不会挂载MSR分区，你也无法在它上面存储数据或删除该分区。进一步地，你也不需要格式化MSR分区，Windows可以直接使用该分区。

DiskPart可以自动设置偏移量参数，来标记MSR分区。大多数场景下，不应该手动设置偏移量参数。EFI的全局唯一标识符（GUID）如下：

e3c9e316-0b5c-4db8-817d-f92df00215ae

3. 创建主分区

在基本磁盘上，可以创建主分区以便存储数据。GPT磁盘支持最多128个分区，其中包含必需的分区与可选的分区。创建的每个主分区都会在GPT头部的GUID分区条目阵列中呈现。如果将包含主分区的基本磁盘转换为动态磁盘，则主分区会变为简单卷，关于这些卷的信息将存储在动态磁盘数据库中，而不再存储于GUID分区条目阵列中。

使用如下命令可以创建主分区：

```
create partition primary size=n
```

其中， n 为以MB为计数单位的空间大小。创建主分区之后，焦点将自动地放置到其上，表示该分区已被选定。刚创建的分区尚未指定盘符与挂载点，但对主分区来讲是需要指定的。要完成主分区的创建，还必须使用支持的文件系统对其进行格式化，比如FAT32或NTFS。

DiskPart可以自动设置偏移量参数，来标记主分区。大多数场景下，不应该手动设置偏移量参数。EFI的全局唯一标识符（GUID）如下：

ebd0a0a2-b9e5-4433-87c0-68b6b72699c7

4. 创建LDM元数据分区与数据分区

在将基本GPT磁盘转换为动态GPT磁盘时，Windows会创建LDM元数据分区与LDM数据分区。LDM元数据分区大小为1MB，用于存储转换操作时所需数据，LDM数据分区用于存储转换时创建的动态卷。

LDM数据分区包含了转换后磁盘上未分配的磁盘空间部分与动态卷（由原来的基本分区转换而来）。比如，如果磁盘包含了一个占据整个磁盘的主引导分区，则转换后的磁盘应该只包含一个LDM数据分区。如果磁盘包含一个引导分区与其他主分区，则转换后将包含两个LDM数据分区（一个用于引导卷，另一个用于其他所有分区）。此外，尽管LDM元数据分区与LDM数据分区不需要盘符与挂载点，但你也可以通过创建主分区（采用上面讲述的方法）来使用这一块磁盘空间。

动态磁盘上LDM元数据分区的GUID为：

5808c8aa-7e8f-42e0-85d2-e1e90434cfb3

动态磁盘上LDM数据分区的GUID为：

af9b60a0-1431:4f62-bc68-3311714a69ad

11.3 管理盘符与挂载点

在对驱动器进行分区后，可以为每个分区分配一个盘符或挂载点，之后对其进行格式化，完成这些工作后，这些分区就可以用于数据存储等用途。通常，可用的盘符从E到Z，A到D一般已经在使用中。在很多系统上，盘符A代表软盘驱动器，盘符B为可移动磁盘驱动器保留，盘符C代表主要的磁盘驱动器，盘符D则通常代表CD-ROM或DVD驱动器。

如果需要更多的分区，可以使用挂载点来创建，并将磁盘挂载到文件系统路径，比如C:\Data。对驱动器路径唯一的限制因素是必须将其挂载到NTFS驱动器上的空文件夹。

11.3.1 分配驱动器盘符或挂载点

要分配盘符或挂载点，可以遵循如下步骤。

- (1) 在命令提示符中键入**diskpart**，调用DiskPart。
- (2) 键入**list volume**命令，列出计算机上当前存在的卷及其盘符分配情况。

注解 只有LIST VOLUME命令可以展示盘符与挂载点分配情况，对计算机上所有分区、逻辑驱动器与卷，该命令都是有效的。这也是为什么使用本命令而不使用LIST PARTITION命令的原因。似乎不完全合乎逻辑，但实际上确实如此。此外，在动态磁盘上为卷分配盘符与挂载点时，这一点也是适用的。

- (3) 遵照如下方法，分别分配盘符或挂载点。

□ 要分配盘符，键入**assign letter=x**，其中，*x*是要使用的盘符，比如：

```
DISKPART> assign letter=f
```

□ 要分配挂载点，键入**assign mount=Path**，其中，*Path*是要用作挂载点的空NTFS文件夹的路径，比如：

```
DISKPART> assign mount=c:\data
```

11.3.2 改变驱动器盘符或挂载点

ASSIGN命令也可以用于改变现有的盘符与挂载点分配信息，其方法与分配盘符类似，也是选定要操作的分区，之后使用ASSIGN为其分配新盘符。之后，DiskPart会改变盘符或挂载点，并要求重新引导计算机，以便所做的修改生效：

```
DiskPart assigned the drive letter, but your computer needs to be rebooted
before the changes take effect.
```

对修改挂载点的情况，DiskPart会报告已经进行了修改，但并不要求重新引导计算机：

```
DiskPart successfully assigned the drive letter or mount point.
```

11.3.3 移除盘符或挂载点

对当前焦点所在分区，可以移除盘符或挂载点，这是通过使用REMOVE命令并遵循如下步骤实现的。

- (1) 在命令提示符中键入**diskpart**，调用DiskPart。

(2) 键入**list volume**命令，列出计算机上当前存在的卷及其盘符分配情况。记住，只有LIST VOLUME命令可以展示盘符与挂载点分配情况，对计算机上所有分区、逻辑驱动器与卷，该命令都是有效的。

(3) 键入**select volume**命令，其后跟随代表指定分区的卷号。这看起来似乎不合逻辑，但这确实是最简单有效的方法。

- (4) 键入**remove**命令，移除选定分区上当前盘符或挂载点。

输入remove命令而不指定任何参数时，会移除其所遇到的第一个盘符或挂载点，并报告如下信息：

```
DiskPart successfully removed the drive letter or mount point.
```

在分区中只包含一个驱动器或挂载点时，上面的方法是有效的。如果分区中包含多个盘符或挂载点，就需要具体指定待移除的盘符与挂载点，分别为**letter=x**与**mount=Path**，比如：


```
DISKPART> remove letter=d
```

或

```
DISKPART> remove mount=D:\Data
```

你也可以移除所有盘符与挂载点，并且指令DiskPart在完成所有移除工作后关闭该卷的所有开放句柄，并卸载该卷。要完成这些任务，可以使用All参数与Dismount参数，如下面一些实例所示。

移除所有盘符与挂载点：

```
DISKPART> remove all
```

移除所有盘符与挂载点，之后卸载相关卷：

```
DISKPART> remove all dismount
```

移除挂载为d:的卷并卸载该卷：

```
DISKPART> remove letter=d dismount
```

注解 在MBR磁盘上，不能移除系统分区或引导分区（以及任何包含了活跃页面文件或崩溃转储（内存转储）的分区）上的盘符；在GPT磁盘上，不能移除EFI、OEM、未识别分区或非数据分区上的盘符，但可以移除可移动驱动器上的盘符。

11.4 格式化分区

格式化的作用是在分区上创建文件系统，并永久删除其上现存的数据。Windows Vista与Windows Server 2008支持的磁盘文件系统包括FAT、FAT32、NTFS。其中，FAT是MS-DOS以及早期Windows支持的文件系统，FAT32是FAT的32位版，NTFS则是Windows NT、Windows 2000、Windows XP、Windows Vista、Windows Server 2003、Windows Server 2008支持的本原文件系统类型。

Windows Vista SP1及其后续版本与Windows Server 2008都支持热插拔的存储介质（使用exFAT卷与NTFS卷）。也就是说，对可移动存储设备，可以有更多的格式化类型。比如，可以将USB闪存设备与类似存储介质格式化为exFAT、FAT16、FAT32、NTFS。exFAT文件系统是FAT系列的下一代文件系统类型，其设计目标是可以在任意相容的操作系统或设备中使用。

11.4.1 使用 FORMAT

与早期的发布版不同的是，Windows Vista与Windows Server 2008中的DiskPart支持一个内部的FORMAT命令。也就是说，在使用DiskPart时，可以对焦点所在分区进行格式化，而不需要退出该工具。实际上，如果键入format，而不提供任何附加参数，则选定的分区会自动进行格式化（使用默认的文件系统与分配单元大小）。然而，典型情况下，并不希望使用默认值进行格式化，而是要指定所需要的格式化参数。DiskPart FORMAT命令的基本语法格式如下：

```
format fs=FileSystem label=Label unit=UnitSize
```

其中，FileSystem设置了要格式化的文件系统类型，Label设置了描述性的文本名，UnitSize则设置了格

式化时每一磁盘簇的分配单元大小（以字节计数）。磁盘簇是由512字节大小的连续扇区所组成的磁盘的一小部分，卷标号是驱动器的文本描述符，适用于驱动器盘符，不适用于挂载点。如果不设置分配单元大小，FORMAT会根据卷实际大小进行选择。有效的分配单元大小包括下面几个。

- ❑ 512。每簇设置为512字节大小。
- ❑ 1024。每簇设置为1024字节大小。
- ❑ 2048。每簇设置为2048字节大小。
- ❑ 4096。每簇设置为4096字节大小。
- ❑ 8192。每簇设置为8192字节大小。
- ❑ 16K。每簇设置为16k字节大小。
- ❑ 32K。每簇设置为32k字节大小。
- ❑ 64K。每簇设置为64k字节大小。

要了解如何使用FORMAT命令，参考如下一些实例。

使用FAT32文件系统格式化选定的卷，并将其标记为AppData：

```
format fs=fat32 label=AppData
```

使用NTFS文件系统格式化选定的卷，并将磁盘簇大小设置为512字节：

```
format fs=ntfs unit=512
```

使用NTFS文件系统格式化选定的卷，并将其标记为AppData：

```
format fs=ntfs label=AppData
```

警告 如果卷上有现存的文件系统，FORMAT并不会弹出提示信息。因此，在选择卷时需要慎重，否则会损毁其上所有现存数据。

有些情况下，你可能需要在格式化某个卷之前卸载它，这可以通过Override参数实现。此外，如果操作的是一个以前进行过格式化并且不存在已知问题的卷，还可以使用Quick参数对其进行快速格式化，而不是彻底格式化。通过快速格式化，FORMAT不需要进行错误检测工作。对大型磁盘，快速格式化会节省一些时间，但不能标记出磁盘的坏扇区并将其隔离。

FORMAT运行时，默认情况下，会显示其格式化进度。通过Nowait参数，你可以在格式化进行过程中强制该命令立即返回。但对于DiskPart控制下的卷，只有在格式化完成后，才能对其进行操作。

DiskPart的FORMAT命令提供了其他几个有用的参数，包括Compress、Revision、Recommended。Compress参数用于将NTFS卷标记为压缩的，表示该卷上创建的文件将使用NTFS压缩功能进行压缩；Revision参数用于将文件系统修订设置为可用的与必要的，比如在格式化CD/DVD介质并希望使用统一磁盘格式（UDF）的特定版本时；Recommended参数用于设置推荐的文件系统类型与修订，而不是使用默认值。

11.4.2 使用 FILESYSTEMS

准备格式化磁盘时，你可以使用FILESYSTEMS命令来显示选定卷上当前文件系统以及该卷支持

的文件系统类型。参考如下示例输出：

Current File System

Type : FAT32
Allocation Unit Size : 32K

File Systems Supported for Formatting

Type : NTFS
Allocation Unit Sizes: 512, 1024, 2048, 4096 (Default), 8192, 16K, 32K, 64K

Type : FAT
Allocation Unit Sizes: 512, 1024, 2048, 4096, 8192, 16K, 32K, 64K (Default)

Type : FAT32 (Default)
Allocation Unit Sizes : 512, 1024, 2048, 4096, 8192, 16K, 32K (Default), 64K

本实例中，当前文件系统类型为FAT32，分配单元大小为32KB。对该卷，可以使用FAT、FAT32、NTFS等文件系统类型进行格式化，分配单元大小则可以设置为上面列出的任意值。如果采用FAT，默认的分配单元大小为64KB；如果采用FAT32，默认的分配单元大小为32KB；如果采用NTFS，默认的分配单元大小为4096字节。

Windows Vista SP1及其后续版本与Windows Server 2008中，对大多数类型的可移动存储设备，exFAT被作为一种附加的格式化选项，而标准的FAT（FAT 12/16）则不再作为一个可用的选项。下面给出的是对可移动存储设备使用FILESYSTEMS命令时获取的输出：

Current File System

Type : FAT32
Allocation Unit Size : 4096

File Systems Supported for Formatting

Type : NTFS
Allocation Unit Sizes: 512, 1024, 2048, 4096 (Default), 8192, 16K, 32K, 64K

Type : FAT32 (Default)
Allocation Unit Sizes: 512, 1024, 2048, 4096 (Default), 8192, 16K, 32K, 64K

Type : exFAT
Allocation Unit Sizes: 512, 1024, 2048, 4096, 8192, 16K, 32K (Default), 64K, 128K, 256K, 1024K, 2048K, 4096K, 8192K, 16384K, 32768K

Windows Vista SP1及其后续版本与Windows Server 2008中，可以使用DiskPart来格式化空白的CD与DVD，下面给出的是对空白DVD使用FILESYSTEMS命令时获取的输出：

Current File System

Type : RAW

Allocation Unit Size : 2048

File Systems Supported for Formatting

Type : UDF [Revision 1.50]
Allocation Unit Sizes : 2048 (Default)

Type : UDF [Revision 2.00]
Allocation Unit Sizes : 2048 (Default)

Type : UDF [Revision 2.01] (Default)
Allocation Unit Sizes : 2048 (Default)

Type : UDF [Revision 2.50]
Allocation Unit Sizes : 2048 (Default)

在这一实例中，类型RAW表示该存储介质尚未格式化。进行格式化后，会发现文件系统类型变为UDF。UDF已经取代了以前使用的CD文件系统（CDFS），尽管你也可以继续使用CDFS，但UDF有很多优势。UDF是一种新兴的文件系统，其操作与使用其他类型的可移动存储类似，比如USB闪存或移动硬盘。你可以直接向其上复制数据——复制与粘贴（或者选中并拖放），而不需要进行刻录等操作。如果磁盘是可重复刻录的，还可以直接选中并删除文件。从系统中移除磁盘后，还可以根据需要再将其插入到CD/DVD驱动器中，就像可移动存储器一样使用。另外，尽管兼容的计算机都可以读取UDF磁盘，但大多数家庭和汽车上的CD/DVD播放器还不能读取UDF磁盘。

Windows Vista与Windows Server 2008支持的UDF版本如下。

- UDF 1.50。一种与Windows 2000及其后续Windows版本兼容的格式，但可能与Windows 98以及苹果计算机不兼容。
- UDF 2.00。一种与Windows XP及其后续Windows版本兼容的格式，但可能与Windows 98、Windows 2000以及苹果计算机不兼容。
- UDF 2.01。默认格式，包含了大多数场景下你所需要利用的重要更新，与Windows XP及其后续Windows版本兼容，但可能与Windows 98、Windows 2000以及苹果计算机不兼容。
- UDF 2.50。Windows Vista中优化的格式，可能与Windows早期版本与苹果计算机不兼容。

在对空白的CD/DVD进行格式化时，可以使用Revision参数来设置需要的UDF版本。要记住必须使用单引号或双引号将版本号包含起来，如下面一些实例所示：

```
format fs=udf revision='1.50'
format fs=udf revision='2.00'
format fs=udf revision='2.01'
format fs=udf revision='2.50'
```

你应该总是使用上面所示的完整的版本标识符。

11.4.3 格式化：一个实例

你可以使用FORMAT命令对焦点所在卷进行格式化，遵循如下步骤。

- (1) 在命令提示符中键入**diskpart**，调用DiskPart。
- (2) 使用**list volume**命令列出计算机上可用卷，注意每个卷的类型、状态与文件系统。分区的卷类

型为Partition, 可移动存储设备的卷类型为Removable, CD设备的卷类型为CD-ROM, CD/DVD设备卷类型为DVD-ROM。如果尚未向可移动存储设备或CD、DVD驱动器中插入存储介质, 则卷状态一般会报告为没有介质。

(3) 键入**select volume**命令, 其后跟随卷的标号, 选择要操作的卷。如果此时再次键入**list volume**命令, 其后带有星号的卷就是所选择的卷。在继续下一步之前, 进行这种确认是有意义的。

(4) 键入**format**命令, 其后跟随需要的参数与参数值, 比如**format fs=ntfs label='secondary data' unit=4096**。

接下来, DiskPart会展示格式化进程。格式化完毕后, 其输出信息类似于如下:

"全部完成DiskPart成功格式化卷。"

如果格式化过程中发生错误导致DiskPart无法完成格式化, 则会报告如下的错误信息:

"DiskPart遇到意外错误。检查系统事件日志获取关于失败的更多信息。"

发生错误的原因可能有多种, 你可以使用FILESYSTEMS命令确认文件系统设置是否正确。有些情况下, 可能需要使用OVERRIDE, 以便在格式化之前强制卸载该卷。

11.5 管理分区

常见的分区管理任务包括将FAT与FAT32分区转换为NTFS、修改卷标、收缩分区、扩展分区、删除分区等, 本节将对这些分区管理任务进行讨论。

11.5.1 将分区或卷转换为 NTFS

如果使用FAT与FAT32创建了分区或卷, 则可以直接将其转换为NTFS格式, 而不需要再进行格式化。所带来的好处是原有的文件与目录结构得以完整保留, 没有数据丢失。要进行这种转换, 可以在命令行中使用CONVERT命令。(注意, DiskPart CONVERT命令用于完全不同的目标。)

1. 转换: 预检查

在实际使用CONVERT命令之前, 应该进行如下一些操作。

- 对分区进行检查, 判断其是否当前正被用作包含操作系统的活跃的引导分区或系统分区。对MBR磁盘上的系统, 可以将活跃的引导分区转换为NTFS。但这样做的前提是CONVERT对该分区具有排他性的访问权限, 这种权限只有在系统启动时才可以具备。因此, 在将活跃的引导分区或系统分区转换为NTFS时, 会看到提示信息询问是否需要在系统下次重启时进行转换。如果选择“是”, 就可以在下次重启时开始转换过程。此外, 要完全转换活跃的引导分区, 通常需要几次重启。
- 检查硬盘驱动器是否具备足够的空闲空间进行转换。要进行转换, 需要一块大小约为该分区或卷使用总空间大小25%的空闲空间。比如, 如果300GB的分区存储了200GB的数据, 则CONVERT需要大约50GB的空闲空间。在运行之前, CONVERT会检查空闲空间大小, 并且在没有足够空闲空间时停止运行。

警告 没有可以将NTFS转换为FAT的工具。将NTFS转换为FAT或FAT32的唯一途径是删除该分区或卷, 之后将其重新创建为FAT或FAT32卷。

2. 进行基本转换

CONVERT命令是在命令提示符中运行的。如果需要对硬盘驱动器格式进行转换，可以使用如下的语法格式：

```
convert volume /FS:NTFS
```

其中，*volume*为卷标识符，其后跟随分号、驱动器路径或卷名。比如，如果需要将D盘转换为NTFS格式，可以使用如下命令：

```
convert D: /FS:NTFS
```

提示 对由FAT或FAT32转换为NTFS格式的卷，主文件表（MFT）的创建位置与直接用NTFS进行格式化的卷中所在位置是不同的，这会导致性能的下降。为对性能进行优化，你可以使用一块指定的转换区域，下面会对其进行讨论。

在转换后的引导卷与系统卷上，CONVERT应用同样的默认安全设置（Windows安装时使用的设置）。在其他卷上，CONVERT需要对安全策略进行适当设置，以便Users组具备对其的访问权限，但不会对Everyone组进行授权。如果需要Everyone组成员访问磁盘上的数据，可以使用/NoSecurity参数删除其上的安全设置，比如：

```
convert D: /FS:NTFS /nosecurity
```

警告 /NoSecurity参数会删除所有安全属性，使得该磁盘上的所有文件与目录都可以被Everyone组成员访问。

CONVERT命令还有几个附加的参数。比如，你可以使用verbose开关（/V参数），用来在转换时获取更为详尽的信息。你也可以使用dismount开关（/X参数），用来在转换之前强制性地卸载分区或卷（如果必要）。在转换之前卸载驱动器的主要原因是确保在分区转换过程中没有应用程序或进程对其进行访问，但不能卸载引导盘或系统盘，这些驱动器只能在系统下次重启时进行转换。

对大多数类型的磁盘，基本转换过程都可以起到良好的作用。但有些时候，基本转换过程不能提供理想的结果。比如，转换之后，磁盘性能降低而不是增强。要克服基本转换过程存在的不足，可以使用/CvtArea参数，该参数的作用将根目录中临近文件的文件名设置为NTFS系统文件的占位符。

3. 使用/CvtArea参数

理想情况下，对那些频繁访问的文件，应该存储在与磁盘起始处相距较近的位置，以便降低寻找与读取该文件所需的时间。由于这一原因，在对磁盘进行格式化时，特定的NTFS系统文件会存储在磁盘起始处。然而，在使用基本转换过程时，Windows无法将新的NTFS系统文件放置到磁盘起始处，因为该区域已经被一些需要保留的其他文件占据。由此导致的结果是，转换后的磁盘性能可能低于原来的FAT或FAT32磁盘。

Windows Vista与Windows Server 2008可以解决这一问题，其方法是使用CvtArea参数，其后跟随要使用的临时文件名。其语法格式为：

```
convert volume /FS:NTFS /CVTAREA: FileName
```

其中，*volume*为盘符，其后跟随分号，*FileName*是一个预先创建的文件，该文件用作临时文件，比如：

```
convert C: /FS:NTFS /CVTAREA:temp.txt
```

通过该命令，主文件表（MFT）与其他NTFS元数据文件都被写入到temp.txt文件，该文件是一个临近的占位符文件。不指定CvtArea参数使用CONVERT时，磁盘起始处的FAT系统文件不会被移动，而会被删除，后面的常规文件则占据这部分区域；指定CvtArea参数使用CONVERT时，CONVERT会查找文件名列表，并在磁盘起始处放置占位符，而不是放置常规文件。在磁盘转换为NTFS之后，CONVERT会删除CvtArea文件，并使用新生成的NTFS系统文件替代它。因而，使用/CvtArea参数可以避免转换后有过多的碎片化的系统文件。

在运行CONVERT之前，可以使用FSUTIL命令创建占位符文件，CONVERT并不会创建这一文件。以达到最优效果，这一文件的大小应该等于1KB乘以文件系统中的文件与目录总数。要确定文件系统中文件与目录总数，最简单的方法是检查磁盘上每一个顶级文件夹的属性。其中记录了文件与文件夹总数，之后计算出所有顶级文件夹包含的文件与目录总数。要做到这一点，可以遵循如下步骤。

- (1) 启动Windows资源管理器，右击驱动器中某顶级文件夹，选择“属性”。
- (2) 观察“包含”条目中文件与文件夹总数，单击“确定”。
- (3) 对每一个顶级文件夹，重复这一过程，之后累加文件与文件夹总数。

获取文件与文件夹总数后，用该数值乘以1KB，所得就是占位符文件大小。比如，如果有1000个文件与文件夹，则占位符文件大小应该为1000KB。要创建占位符文件，可以键入如下命令：

```
fsutil file createnew FileName ByteSize
```

其中，*FileName*是要创建的文件名，*ByteSize*是该文件的字节总数。比如，要创建一个名为Temp.txt的1000KB大小的文件，就应该使用如下命令（每KB包含1024字节）：

```
fsutil file createnew temp.txt 1024000
```

注解 CONVERT会使用NTFS元数据重写这一文件，转换后，此文件中未使用空间将被释放。

11.5.2 改变或删除卷标

卷标是盘符之外的文本描述符，用于简要描述该卷的大致用途。对FAT或FAT32卷，卷标最多可以包含11个字符，也可以包含空格。对NTFS卷，卷标可以包含32个字符。此外，对于FAT与FAT32卷标不允许使用的特殊字符，包括*、/、\、[、]、:、;、|、=、,、.、+、"、?、<、>，NTFS卷标也可以使用。

在多种工具中访问磁盘时，都会显示卷标，比如Windows资源管理器。在命令提示符中，可以使用LABEL命令修改或删除卷标。在DiskPart命令中，并没有可用于改变或删除卷标的LABEL命令，但在DiskPart的FORMAT命令中，有一个Label参数。

要修改卷标，可以使用如下的语法格式：

```
label drive: label
```

其中，*drive*为驱动器盘符，其后跟随分号，*label*则为分配的文本描述符，比如：

```
label f: AppData
```

注解 在操作卷标时，VOL是一个有用的命令，可以列出当前卷名（如果有）。

11.5.3 压缩分区或卷

在基本磁盘上压缩分区与在动态磁盘上压缩卷的技术是一样的。如果创建了一个过于庞大的卷，

有时候可能需要减少其容量，以便为其他卷腾出空间。对卷进行压缩时，可以不区分针对的是基本磁盘还是动态磁盘。压缩卷的过程中，实际上是从该卷上移除未使用的空间。

在对卷进行压缩时，有几个限制因素。比如，只能对NTFS卷进行压缩，而不能对FAT或FAT32卷进行压缩；可以对此前未格式化的卷进行压缩，但不能对带区卷进行压缩。

要对分区进行压缩，可以遵循如下步骤。

- (1) 在命令提示符中键入**diskpart**，调用DiskPart。
- (2) 键入**list disk**，列出计算机上所有磁盘，并检查空闲空间。
- (3) 选择待操作的磁盘，比如，要选择disk 2，则应键入**select disk 2**。
- (4) 键入**list partition**，列出选定磁盘上的分区。
- (5) 选择待操作的分区，比如，键入**select partition 2**。
- (6) 键入**shrink querymax**，确定可以从该卷释放的最大空闲空间。

提示 大多数场景下，并不会按最大空闲空间来释放，而是要为其保留足够的空闲空间，以便在磁盘进行读写操作时保持较好性能。建议保留至少10%的空闲空间。在对系统卷、引导卷以及包含页面文件与影拷贝的卷进行操作时，应该保留更多的空闲空间。

(7) 键入**shrink desired=*n***，对分区进行压缩。其中，*n*为要移除的磁盘空间总量（以MB计数），比如：

```
DISKPART> shrink desired=1000
```

注解 如果空间总量大小在柱面边界附近，会导致移除空间总量或多或少的些微变化。

如果有足够的空闲空间，则DiskPart会根据指定的总量来释放卷空间，否则会根据该卷上可用的最大空闲空间量进行释放。

你也可以以其他方式使用SHRINK。选定一个卷后，只需键入**shrink**，而不带任何参数，则DiskPart会根据该卷上可用的最大空闲空间量进行释放。你也可以使用Nowait参数，该参数的作用是使得DiskPart立即返回（尽管磁盘压缩过程正在进行），但是不应该在DiskPart中操作该卷，直至压缩操作完成。

此外，你也可以使用Minimum参数来指定要移除的最小磁盘空间总量（以MB计数）。在指定这一参数后，Diskpart或者压缩至少这样大小的磁盘空间，或者不移除任何磁盘空间。下面的实例中，将指定的压缩值设置为2400MB，最小压缩值设置为1200MB：

```
shrink desired=2400 minimum=1200
```

根据上述命令，Diskpart将尝试至少压缩1200MB的磁盘空间。如果做不到这一点，则Diskpart不会从磁盘上移除任何磁盘空间。

11.5.4 扩展分区或卷

在基本磁盘上扩展分区与在动态磁盘上扩展卷的技术是一样的。如果创建了一个过小的卷，有时候就可能需要扩展它。对卷进行扩展时，可以不区分针对的是基本磁盘还是动态磁盘。扩展卷的过程中，实际上是转换未分配的空间，并将其添加到现存卷。对动态磁盘上的跨区卷，空间可以来自任意

可用的动态磁盘，而不仅仅是创建该卷时所在磁盘。因此，通过这种扩展，可以将多个动态磁盘上的空闲空间区域结合在一起，并使用这些区域来提高现存卷的总容量。

与压缩卷时类似，在对卷进行扩展时，也有几个限制因素。比如，只能对NTFS卷进行扩展，而不能对FAT或FAT32卷进行压缩；可以对此前未格式化的卷进行压缩，但不能对带区卷进行扩展；不管其配置怎样，都不能扩展系统卷或引导卷。此外，在基本磁盘上，扩展的空闲空间必须与选定的分区或卷在同一磁盘上，并且紧随在该分区或卷之后。也就是说，空闲空间必须从下一个扇区偏移量开始。

要对卷进行扩展，可以遵循如下步骤。

- (1) 在命令提示符中键入**diskpart**，调用DiskPart。
- (2) 键入**list disk**，列出计算机上所有磁盘，并检查空闲空间。
- (3) 选择待操作的磁盘，比如，要选择disk 2，则应键入**select disk 2**。
- (4) 键入**list volume**，列出选定磁盘上的卷。
- (5) 选择待操作的卷，比如，键入**select volume 6**。
- (6) 键入**extend size=n**，其中，*n*为要增加的空闲空间总量（以MB计数），比如：

```
DISKPART> extend size=1000
```

注解 如果size大小在柱面边界附近，会导致增加空间总量或多或少的些微变化。如果没有指定size，则该卷会扩展到填充磁盘上未分配空间。

11.5.5 删除分区

如果需要改变一个已经完全分配的驱动器的配置，可能就需要先删除现存的分区。删除分区会移除其上相关联的文件系统，其上所有数据将丢失。因此，在删除分区之前，应该对其中包含的文件与目录进行备份。

在基本磁盘上，你可以使用DELETE PARTITION命令删除焦点所在分区，但不能使用这一命令删除系统分区或引导分区，也不能删除包含活跃页面文件或崩溃转储文件（内存转储）的分区。要了解如何使用DELETE PARTITION，参考如下实例。

- (1) 在命令提示符中键入**diskpart**，调用DiskPart。
- (2) 键入**list disk**，按Enter键，列出计算机上磁盘。键入**select disk**，其后跟随磁盘编号，选定要操作的基本磁盘。
- (3) 键入**list partition**，列出选定磁盘上的分区。
- (4) 键入**select partition**，其后跟随分区编号。选定要删除的分区，之后键入**delete partition**，删除该分区。

更多信息 DiskPart只可以删除已知的数据分区。如果你确信了解自己所做操作及其后果，也可以为DELETE PARTITION命令添加Override参数，来扩大可删除分区的范围。

要注意的是，删除基本磁盘上分区与删除动态磁盘上卷的技术是不同的。不要试图使用DELETE PARTITION命令删除动态磁盘上卷，而应该使用DELETE VOLUME命令，该命令将在12.2.4节讲述。

操作动态磁盘时，创建的是卷，而不是分区。简单地说，卷是可以直接用于存储数据的磁盘部分。尽管创建卷与创建分区有很多相似之处，但卷有很多附加的功能。创建卷时，可以完成如下一些任务。

- 在单一驱动器上创建一个卷，称为简单卷。
- 对卷进行扩展，使其占据磁盘上所有空闲空间，这一过程创建了一个扩展卷。
- 创建一个跨越多个驱动器的卷，称为跨区卷。
- 配置RAID(独立磁盘冗余阵列)，Windows Server 2008支持RAID-0、RAID-1与RAID-5，Windows Vista则仅支持RAID-0。

由于卷与RAID阵列是在动态驱动器上创建的，因此只能由Windows 2000、Windows XP、Windows Vista、Windows Server 2003和Windows Server 2008访问。如果将计算机设置为包含早期Windows版本的双引导系统，则其上的动态驱动器对早期Windows版本是不可用的。然而，在网络上，可以像访问其他驱动器一样访问动态驱动器。这意味着运行早期Windows版本的计算机可以通过网络访问动态驱动器。

12.1 获取卷信息与状态

在使用DiskPart并希望检查分区或卷的状态时，可以使用LIST VOLUME命令。如下面实例所示，LIST VOLUME命令可以列出计算机上所有卷、分区、逻辑驱动器的统计信息：

```
DISKPART> list volume
```

Volume ###	Ltr	Label	Fs	Type	Size	Status	Info
Volume 0	N			Removable	0 B	No Media	
Volume 1	F	Blank Disc	CDUDF	DVD-ROM	2048 B	Healthy	
Volume 2	J			Removable	0 B	No Media	
Volume 3	I			Removable	0 B	No Media	
Volume 4	L			Removable	0 B	No Media	
Volume 5	G			DVD-ROM	0 B	No Media	
Volume 6	E	Recovery	NTFS	Simple	9 GB	Healthy	
Volume 7	C		NTFS	Mirrored	457 GB	Healthy	System
Volume 8	D		NTFS	Mirrored	457 GB	Healthy	
Volume 9	O		NTFS	RAID-5	466 GB	Healthy	
Volume 10	Q		NTFS	RAID-5	466 GB	Healthy	
Volume 11	P		NTFS	RAID-5	466 GB	Healthy	
Volume 12	K		NTFS	Simple	477 GB	Healthy	

从上面的输出信息可以看出，LIST VOLUME命令可以展示如下一些要素。

- Volume ###。卷编号，你可以通过select volume n命令来选定待操作的卷。
- Ltr。卷的驱动器盘符。
- Label。卷标。
- Fs。文件系统类型，比如CDUFD、FAT、FAT32或NTFS。
- Type。布局类型，对于动态磁盘，卷布局类型会报告卷的配置为简单、跨区、镜像、条带或RAID-5。
- Size。卷的总存储容量。
- Status。卷的状态，展示为正常、失败冗余等。
- Info。提供了关于该卷的一些附加信息，比如该卷是否为系统卷等。

卷状态是重要的一个统计信息。在安装新卷或进行故障排除时，理解卷状态是有用的。表12-1总结了一些卷状态值，主要是与动态卷相关的。在DiskPart中，可以使用如下的命令执行相关任务。ONLINE命令，用于使得某个卷联机；RECOVER命令，用于在磁盘上进行恢复与再同步；REPAIR，用于修复失效的RAID-5成员；RESCAN，用于重新扫描计算机的磁盘与卷。

表12-1 理解与解决卷状态问题

状 态	描 述	解 决
数据不完整	在外部磁盘上的跨区卷不完整，你可能忘记从跨区卷中添加其他磁盘	遍历包含跨区卷剩余部分的磁盘，一次性导入所有磁盘
数据非冗余	在外部磁盘上的容错卷不完整（非冗余），你可能忘记从镜像或RAID-5集中添加其他磁盘	添加剩余的磁盘，一次性导入所有磁盘
失败	磁盘错误状态，该磁盘不可访问或者已损坏	确保相关的动态磁盘联机。如果有必要，重新扫描卷。或将该卷联机。你也可以尝试恢复该卷
失败冗余	磁盘错误状态，镜像或RAID-5中的某个磁盘状态脱机	确保相关的动态磁盘状态为联机。如果有必要，重新扫描卷。或将该卷联机。你也可以尝试恢复该卷。如果卷无法联机，则可能需要替换失效的镜像或修复失效的RAID-5卷
格式化	临时状态，表示该卷正处于格式化过程中	格式化过程会显示完成的百分比，成功格式化之后，卷状态会设置为良好
良好	正常的卷状态	卷没有任何已知的问题
良好（有风险）	Windows在读、写动态卷所在物理磁盘时出现问题，在Windows遇到错误时会呈现这一状态	确保相关的动态磁盘状态为联机。如果有必要，重新扫描卷，或将该卷联机。你也可以尝试恢复该卷。如果卷持续呈现这一状态或者周期性呈现这一状态，则说明该卷可能已经失效，应该对磁盘上所有数据进行备份
良好（未知分区）	Windows无法识别分区，在分区来自不同的操作系统，或者是手工创建的用于存储系统文件的分区时，就可能呈现这一状态	不需要纠正措施

(续)

状 态	描 述	解 决
初始化	临时状态, 表示该磁盘正处于初始化过程中	驱动器状态在几秒之后会变化
正在重新生成	临时状态, 表示镜像卷正被添加或导入, 或者RAID-5的数据与奇偶信息正被重建	重建过程会显示完成的百分比。成功重建之后, 卷状态会设置为良好
重新同步	临时状态, 表明镜像集正被再同步	再同步过程会显示完成的百分比。成功完成之后, 卷状态会设置为良好
过期数据	容错的外部磁盘上的数据失去同步	重新扫描磁盘, 或者重新同步镜像卷或RAID-5卷, 之后检查状态时应该发现已经显示为新状态, 比如失败冗余
未知	卷不能被访问, 可能包含损坏的引导扇区	该卷可能中了引导扇区病毒, 用最新病毒库对其进行检测, 重新扫描磁盘或重启计算机, 之后检查状态

12.2 创建并管理简单卷

对动态磁盘, 可以使用DiskPart创建简单卷, 且简单卷是最基本的动态卷类型。与分区不同的是, 简单卷可以遍布整个磁盘。当然, 你也可以根据所配置的计算机的实际需要为简单卷指定适当的大小。

12.2.1 创建简单卷

在向磁盘添加简单卷之前, 应该对磁盘上空闲空间总量进行评估, 并检查当前卷配置。你可以遵循如下步骤完成这些任务。

- (1) 在命令提示符中键入**diskpart**, 调用DiskPart。
- (2) 列出计算机上的磁盘, 并检查空闲空间, 使用如下命令:

Disk ###	Status	Size	Free	Dyn	Gpt
-----	-----	-----	-----	---	---
Disk 0	Online	372 GB	0 B		
Disk 1	Online	329 GB	204 GB	*	
Disk 2	Online	337 GB	37 GB	*	

这一实例中, disk 1与disk 2被格式化为动态磁盘(由Dyn列的*可以看出), 分区格式为MBR(由Gpt列为空白可以看出)。Disk 1有204GB的可用空闲空间, disk 2有37GB的可用空闲空间。

确定了待操作的磁盘后, 就可以使用如下命令创建简单卷:

```
create volume simple size=n disk=n
```

其中, *size=n* 设置了卷的大小(以MB计数), *disk=n* 指定了要操作的磁盘编号。

创建卷之后, 焦点将自动地被放置到其上, 表示该卷已被选定。但该卷尚无盘符或挂载点, 必须使用ASSIGN命令进行分配。要完成最后的创建工作, 还必须使用FORMAT命令对其进行格式化。对于卷与分区, 这些任务的执行方式都是一致的。要了解更多信息, 参阅11.3.1小节与11.3.4小节。

12.2.2 扩展简单卷

如果某简单卷需要更多磁盘空间，有两种方式可以对其进行扩展。一种是在同一磁盘上对其进行扩展，所得的卷叫做扩展卷；一种是在其他磁盘上进行扩展，所得的卷叫做跨区卷。对这两种情况，卷都必须格式化为NTFS格式。

要对简单卷进行扩展，可以遵循如下步骤。

- (1) 在命令提示符中键入**diskpart**，调用DiskPart。
- (2) 列出计算机上的磁盘，并检查空闲空间，使用如下命令：

```
DISKPART> list disk
```

- (3) 列出计算机上的卷：

```
DISKPART> list volume
```

- (4) 选定待扩展的卷，比如volume 5：

```
DISKPART> select volume 5
```

- (5) 对卷进行扩展。

- 要在当前磁盘上对卷进行扩展，使用如下命令：

```
DISKPART> extend size=n
```

其中，*size=n*设置了要增加的空间大小（以MB计数）。比如，使用如下命令，可以将选定的卷扩展1004MB：

```
DISKPART> extend size=1004 disk=2
```

注解 *size*大小在最近柱面边界附近，这会导致最终扩展的磁盘空间大小有或多或少的些微变化。

- 要在另外的动态磁盘上对卷进行扩展，使用如下命令：

```
DISKPART> extend size=n disk=n
```

其中，*size=n*设置了要增加的空间大小（以MB计数），*disk=n*指定了卷应该扩展到的磁盘。比如，如果卷在disk 0上，现在需要将其扩展到disk 1，可以使用如下命令：

```
DISKPART> extend size=2008 disk=1
```

这里，将disk 0的卷扩展到disk 1上，在disk 1上的扩展区域大小为2008MB。

警告 扩展卷集时，有很多限制因素。你不能扩展引导卷或系统卷，不能扩展使用镜像或条带卷（RAID-0、RAID-1和RAID-5），也不能将卷扩展到32块以上的磁盘。此外，FAT卷与FAT32卷不能直接扩展。在扩展之前，必须先将其转换为NTFS格式。

12.2.3 将动态磁盘联机

动态磁盘有比基本磁盘丰富得多的功能。你可以很容易地解决其上存在的错误，并返回到已经脱机的驱动器。你也可以检查驱动器配置的变化，并在不同计算机之间迁移与导入磁盘。

如第10章中所讨论的，LIST DISK命令可以展示系统中每个可用磁盘的状态。如果某动态磁盘状

态显示为联机（错误）或脱机，通常可以使用ONLINE命令纠正这一问题；DiskPart命令则可以用于指定待操作的磁盘，比如先键入**select disk 0**，之后键入**online**。如果驱动器状态没有改变，则可能需要重新引导计算机。如果重新引导计算机后仍不能解决问题，则需要检查驱动器及其控制器、电缆、电源等，确保相关硬件都已经正确连接。通过ONLINE命令，还可以重新同步镜像卷或RAID-5卷。

如果驱动器配置已经改变，或者新的磁盘被添加到计算机中，则可以使用RESCAN命令重新扫描计算机上所有驱动器，并检查驱动器配置的更新。有时候，重新扫描还可以解决驱动器状态显示为不可读等问题。

如果将动态磁盘从某计算机移动到其他计算机，则该磁盘可能会被标记为外部磁盘。磁盘可能标记为外部的另一种情况是该磁盘已经失效，但又试图将其联机。如果需要使用DiskPart将磁盘联机，可以先使用**select disk 0**选定待操作的磁盘，之后使用**import**命令。

12.2.4 删除卷

在动态磁盘上，不应该使用DELETE PARTITION命令，因为该命令将删除磁盘上所有动态卷。如果需要删除动态磁盘上焦点所在卷，可以使用DELETE VOLUME命令。与DELETE PARTITION命令类似，你也不能使用DELETE VOLUME命令删除系统卷或引导卷，也不能删除包含活跃页面文件或崩溃转储文件（内存转储）的卷。

要了解如何使用DELETE VOLUME命令，可以参考如下实例。

(1) 在命令提示符中键入**diskpart**，调用DiskPart。

(2) 列出计算机上的卷，使用如下命令：

```
DISKPART> list volume
```

(3) 选定待删除的卷并删除该卷：

```
DISKPART> select volume 5  
DISKPART> delete volume
```

更多信息 默认情况下，DiskPart只可以删除已知的数据卷。至于分区，你可以在DELETE VOLUME命令中使用Override参数绕过这种限制。

12.3 通过动态磁盘上的 RAID 提供容错功能

通过RAID，可以为重要数据提供增强的保护功能，来放置驱动器失效导致的数据丢失。RAID可以在硬件层面实现，也可以在软件层面实现。硬件RAID需要使用硬件销售商提供的工具实现与管理，软件RAID则可以使用操作系统本身实现与管理。

在动态磁盘上，Windows Vista支持RAID-0，Windows Server 2008则支持下面3个层面的RAID。

- **RAID-0**。磁盘分割。RAID-0包含两个或多个卷，每个卷都分布在不同的驱动器，并配置为一个条带集。数据被划分为多个块，称为条带，多块数据被顺序写入条带集所在的所有驱动器。RAID-0提供了增快的速度与增强的性能，但没有提供容错功能。
- **RAID-1**。磁盘镜像与双控。RAID-1中包含两个卷，分布在两个驱动器上，所做的配置是等同的。数据同时写入到两个驱动器。如果某个驱动器失效，由于另一个驱动器上也包含了同样

的数据，因而不会造成数据丢失。RAID-1提供了数据冗余以及比带奇偶校验的磁盘分割（RAID-5）更好的写性能。

- RAID-5。带奇偶校验的磁盘分割。RAID-5包含3块或更多的卷，每个卷分布在不同的磁盘上，以便创建带奇偶错误校验的条带集。失效的时候，数据可以得以恢复。RAID-5提供了容错功能，并且比镜像技术有更低的开销与更高的性能。

12.3.1 实现 RAID-0：磁盘分割

RAID-0，即磁盘分割，包含了两个或更多的卷，每个卷位于不同的驱动器，配置为条带集。写入到条带集的数据被分割为称为条带的数据块，且这些条带按顺序写入条带集中的所有驱动器。此外，尽管理论上讲，可以将条带集中的卷分布在最多32块驱动器上。但大多数场景下，条带集中包含2个到5个卷可以提供最好的性能提升，超过这一数值后，提升的性能将大幅下降。

1. 使用RAID-0

使用RAID-0关键的一个原因就是速度的提升。由于可以使用多个驱动器磁头在多个磁盘上访问数据，因此，读性能可以大幅提升。然而，这种方式也提高了灾难性故障的可能性。如果条带集中的某个硬盘驱动器失效，则条带集将无法继续使用，并且其上所有数据将丢失。如果需要恢复，你可能需要重建条带集并从备份中恢复数据。数据备份与恢复在*Windows Server 2008 Administrator's Pocket Consultant*（Microsoft Press，2008）一书的第16章进行了讨论。

创建条带集时，应该记住如下一些要点。

- 引导卷与系统卷不可以作为条带集的一部分，对这些卷不能使用磁盘分割技术。
- 条带集的总容量以最小卷容量为基础。有鉴于此，在创建条带集时，应该使用大小大致相等的卷。
- 通过使用不同磁盘控制器上的磁盘，可以使得性能提升最大化。因为这种方式可以使得系统同时访问多个驱动器。

在使用RAID-0的磁盘上，运行LIST DISK命令或DETAIL DISK命令，会发现卷类型显示为STRIPED。如果在条带卷上使用DETAIL VOLUME命令，DiskPart会展示条带集中所有简单卷。

在条带卷被破坏时，磁盘状态会显示为丢失。你可以在余下的驱动器上使用DETAIL DISK命令，此时显示的状态应为Failed，表示磁盘冗余已经失效。如果发现存在Failed状态，但不知道还有哪些其他磁盘是该条带卷的一部分，则可以在计算机上的每块其他磁盘上运行DETAIL DISK命令，以便追踪出现故障的磁盘——该磁盘状态应该显示为丢失。

典型情况下，要修复条带集，需要移除失效磁盘、使用新磁盘对其进行替代、将新磁盘配置为条带集的一部分等几个步骤。为此，你可以运行DiskPart，选择新磁盘，之后运行CONVERT DYNAMIC命令转换磁盘类型，之后对新磁盘进行格式化，并为其分配盘符。完成上述工作后，需要使用DiskPart移除驱动器（该驱动器是损坏的条带集的一部分）上的卷，并使用CREATE VOLUME STRIPE命令创建一个新的条带集。完成上述过程后，你可以选择条带卷，并使用LIST VOLUME命令查看其状态。如果一切正常，状态应该显示为良好。

警告 移除卷后，驱动器上所有数据将丢失，你只能从备份中重建数据。如果没有对数据进行备份，则不要重写驱动器。如果一旦数据已经丢失，也可以尝试使用第三方恢复工具恢复一部分数据。

2. 配置条带集

要实现RAID-0, 可以遵循如下步骤。

(1) 在命令提示符中键入**diskpart**, 调用DiskPart。

(2) 列出计算机上的磁盘, 并检查空闲空间, 并确保待操作磁盘都已经配置为动态磁盘, 使用如下命令:

```
DISKPART> list disk
```

(3) 使用下面的命令创建条带集:

```
DISKPART> create volume stripe size=n disk=n,n,n,...
```

其中, *size=n*表示卷在每个磁盘上使用的空间总量(以MB计数)。如果没有指定*size*, DiskPart将使用最小磁盘上的所有剩余空间, 并在其他磁盘上也使用这一数值。*disk=n,n,n,...*表示的是磁盘编号, 卷将被分割在这些磁盘上, 你需要使用至少两块动态磁盘。

参考如下实例。

使用最小磁盘上的所有可用空间在disk 0、disk 1、disk 2上创建条带卷, 之后在所有剩余磁盘上使用同样的空间总量:

```
create volume stripe disk=0,1,2
```

使用80GB (81920MB) 的空间在disk 0、disk 1、disk 2上创建条带卷:

```
create volume stripe size=81920 disk=0,1,2
```

注解 将硬件RAID与软件RAID结合在一起使用时, 你可能需要将卷或分区的起点与最临近的对其边界对齐, 来提高性能。为此, 你可以使用Align参数, 语法格式为align=n。其中, n为从磁盘起始处到最近的对其边界的KB数, 比如align=64。要记住的是, 为与最近的对其边界协调, 此值可能会有或大或小的变化, 并改变分区或逻辑驱动器的最终大小。

12.3.2 实现 RAID-1: 磁盘镜像与双控

12

RAID-1, 也称为磁盘镜像。这种技术中, 在两个不同的驱动器上使用等同大小的卷, 以便创建冗余的数据集。相互镜像的驱动器包含了完全等同的信息, 也就是说, 你可以只从主镜像中读取数据, 但可以向两个驱动器中写入数据。由于有必要两次进行写数据操作, 通常每个镜像驱动器有自己的磁盘控制器, 以便同时向两个驱动器中写入数据。使用两个磁盘控制器时, 驱动器称为双控的。因而, 磁盘镜像与磁盘双控的区别在于是使用一个磁盘控制器还是两个(本章余下部分中, 将不对其进行区分)。

1. 使用RAID-1

使用磁盘镜像关键一个原因是, 在一块磁盘失效后, 另外一块磁盘可以自动地用于数据读写。你也可以使用正常工作的驱动器, 在同一块磁盘或其他磁盘上重建失效驱动器数据。在对失效驱动器进行修复之前, 你需要分离镜像。要了解如何做到这些, 可以参考12.4节。

磁盘镜像的不足之处是带来了额外的开销, 即磁盘镜像会将有效存储空间减半。比如, 如果需要镜像一个750GB的驱动器, 就需要另一个750GB的驱动器。也就是说, 使用磁盘镜像时, 如果需要存储750GB的数据, 就需要1500GB的存储空间。

注解 与磁盘分割不同的是，磁盘镜像可以对任意类型的简单卷进行镜像，包括引导卷与系统卷。

在使用RAID-1的磁盘上，运行LIST DISK命令或DETAIL DISK命令，会发现卷类型显示为Mirrored。如果在镜像卷上使用DETAIL VOLUME命令，DiskPart会展示镜像集中两个卷。

在镜像卷被破坏时，磁盘状态会显示为丢失。你可以在余下的磁盘上使用DETAIL DISK命令，此时显示的状态应为失败冗余，表示磁盘冗余已经失效。如果发现存在失败冗余状态，但不知道还有哪些其他磁盘是该镜像集的一部分，则可以在计算机上的每块其他磁盘上运行DETAIL DISK命令，以便追踪出现故障的磁盘——该磁盘状态应该显示为丢失。

典型情况下，要修复镜像集，需要移除失效磁盘、使用新磁盘对其进行替代、将新磁盘配置为镜像集的一部分等几个步骤。为此，你可以运行DiskPart，选择新磁盘，之后运行CONVERT DYNAMIC命令转换磁盘类型。之后，你需要使用BREAK DISK命令打破现存的镜像，再之后使用ADD DISK命令将新磁盘指定为添加到新镜像集中的磁盘。完成上述过程后，你可以选择镜像卷，并使用LIST VOLUME命令查看其状态。如果一切正常，状态应该显示为良好。

2. 配置磁盘镜像与双控

要创建镜像集，可以先选择想要镜像的简单卷，之后添加一个磁盘作为镜像集中的另一个驱动器。从驱动器中至少应该包含与选定卷大小相等的未分配空间，且遵循如下步骤。

(1) 在命令提示符中键入diskpart，调用DiskPart。

(2) 列出计算机上的磁盘，并检查空闲空间，并确保待操作磁盘都已经配置为动态磁盘，使用如下命令：

```
DISKPART> list disk
```

(3) 选择需要进行镜像的磁盘，下面的命令选定disk 0进行镜像：

```
DISKPART> select disk 0
```

(4) 添加一个磁盘，该磁盘在镜像集中用作第二块硬盘。下面的实例中，添加的是disk 1：

```
DISKPART> add disk=1
```

进行上述操作后，操作系统开始镜像创建过程，两个卷的状态都显示为再同步。如果希望DiskPart等待，直至卷完成同步才返回，则可以使用Wait参数。此外，与RAID-0中的磁盘分割类似，你也可以使用Align参数，使得分区或卷的起点与最近邻的对齐边界对齐。

12.3.3 实现 RAID-5：带奇偶校验的磁盘分割

RAID-5，也称为带奇偶校验的磁盘分割。这种技术中，至少要使用3个硬盘驱动器，以便使用等同大小的卷实现容错功能。使用RAID-5关键的一个原因是，可以防止计算机的单一磁盘失效。如果两个磁盘实现，则奇偶校验信息不足以恢复数据，此时需要从备份数据中重建条带集。

1. 使用RAID-5

你可以把RAID-5理解为增强版的RAID-0，不仅通过磁盘分割提高了性能，还通过奇偶校验增强了容错功能。因而，与RAID-0不同的是，单一驱动器的失效将不再使得整个条带集失效，而是继续使用条带集中余下的正常卷继续进行工作。并且，这些余下的正常卷还可以用于在新磁盘（或者12.4中讨论的恢复磁盘）上重建条带集。

警告 引导卷或系统卷不能作为条带集的一部分，不要在这些卷上使用带奇偶校验的磁盘分割技术。

在使用RAID-5的磁盘上，运行LIST DISK命令或DETAIL DISK命令，会发现卷类型显示为RAID-5。如果在RAID-5卷上使用DETAIL VOLUME命令，DiskPart会展示其中的所有卷。

在RAID-5卷被破坏时，磁盘状态会显示为丢失。你可以在余下的磁盘上使用DETAIL DISK命令，此时显示的状态应为失败冗余，表示磁盘冗余已经失效。如果发现存在失败冗余状态，但不知道还有哪些其他磁盘是该镜像集的一部分，则可以在计算机上的每块其他磁盘上运行DETAIL DISK命令，以便追踪出现故障的磁盘——该磁盘状态应该显示为丢失。

典型情况下，要修复RAID-5集，需要移除失效磁盘、使用新磁盘对其进行替代、将新磁盘配置为RAID-5集的一部分等几个步骤。为此，你可以运行DiskPart，选择新磁盘，之后运行CONVERT DYNAMIC命令转换磁盘类型。之后，你需要使用SELECT DISK命令选定RAID-5卷，再之后使用REPAIR DISK命令将新磁盘指定为待使用的磁盘，这将重建RAID-5集并使得新磁盘成为该集的一部分。完成上述过程后，你可以选择RAID-5卷，并使用LIST VOLUME命令查看其状态。如果一切正常，状态应该显示为良好。

2. 配置带奇偶校验的条带集

要创建RAID-5，可以先选择3个动态磁盘。这些磁盘上应该具备足够的未分配空间，以便创建指定大小的RAID-5集。遵循如下步骤。

(1) 在命令提示符中键入**diskpart**，调用DiskPart。

(2) 列出计算机上的磁盘，并检查空闲空间，并确保待操作磁盘都已经配置为动态磁盘。使用如下命令：

```
DISKPART> list disk
```

(3) 创建RAID-5集，如下：

```
DISKPART> create volume raid size=n disk=n,n,n,...
```

其中，*size=n*表示卷在每个磁盘上使用的空间总量（以MB计数）。如果没有指定*size*，DiskPart将使用最小磁盘上的所有剩余空间，并在其他磁盘上也使用这一数值。*disk=n,n,n,...*表示的是磁盘编号，RAID-5集将被分割在这些磁盘上，你需要使用至少3块动态磁盘。

参考如下实例。

使用最小磁盘上的所有可用空间在disk 2、disk 3、disk 4上创建RAID-5集，之后在剩余磁盘上使用同样的空间总量：

```
create volume raid disk=2,3,4
```

使用80GB（81920MB）的空间在disk 2、disk 3、disk 4上创建RAID-5卷：

```
create volume raid size=81920 disk=2,3,4
```

注解 与磁盘分割与磁盘镜像类似，在RAID-5中，也可以使用Align参数使得卷、分区与最近邻对齐边界对齐。但在创建RAID-5之后，不能再对其进行扩展。有鉴于此，在创建RAID-5之前，应该认真规划它。

12.4 管理 RAID 并从失效中恢复

在管理镜像驱动器与条带集时，所采用的方法不同于管理其他类型的卷。如果某个镜像驱动器或条带集失效，必须以特定方式进行恢复。如果需要终止使用磁盘镜像，必须分离镜像集。如果需要终止RAID-5，必须删除整个卷集。

12.4.1 分离镜像集

在需要终止使用磁盘镜像或重建镜像集时，都需要分离镜像集。如果不再需要对驱动器进行镜像，你可以分离镜像集，并使用那些只存在于一个驱动器上的数据，另外驱动器上的空间则可用于其他用途。如果镜像集中某个驱动器失效，磁盘操作会使用余下的磁盘驱动器继续进行。要修复镜像，必须首先分离镜像集，之后对其进行重建。

提示 尽管分离镜像集并不会删除镜像集中的数据，但你还是应该在分离镜像集之前对其中数据进行备份，这将确保出现故障时可以对数据进行恢复。

如果需要分离镜像集，可以遵循如下步骤。

(1) 在命令提示符中键入**diskpart**，调用DiskPart。

(2) 列出计算机上的磁盘，并确定哪些磁盘是镜像集的一部分，使用如下命令：

```
DISKPART> list disk
```

(3) 在指定的磁盘上分离镜像，进行分离之后，该磁盘将不再包含驱动器盘符与挂载点。比如，如果disk 0与disk 1是镜像的，现在希望用户只继续使用disk 0，则可以使用如下命令分离镜像：

```
DISKPART> break disk=1
```

分离镜像之后，仍然拥有两个包含了相同信息的驱动器，但只有disk 0有可用的盘符或挂载点。如果需要在分离镜像之后丢弃第2个磁盘上的重复信息，并将其上所有磁盘空间作为未分配空间，则可以使用**Nokeep**参数，如下：

```
DISKPART> break disk=1 nokeep
```

12.4.2 重新同步与修复镜像集

镜像集中某个驱动器失效后，在恢复镜像之前，需要对镜像集进行修复。方法是分离镜像集，之后在新驱动器（或者已恢复的失效驱动器）上重建镜像。有时候，所面临的并不是直接失效，而是数据不同步的情况。比如，镜像集中某个驱动器由于某些原因脱机，从而导致数据只能写入一个驱动器中。

要恢复镜像集，你需要将镜像集中的两个驱动器都联机，所采取的纠正措施则依赖于失效驱动器的具体状态，如下所示。

- ❑ 如果驱动器状态为丢失、脱机或良好（有风险），检查该驱动器是否供电，以及物理连接是否正确。之后，启动DiskPart，使用**RESCAN**命令检测该卷，再之后使用**ONLINE**命令将该磁盘联机。如果一切正常，则驱动器状态应该逐渐从正在重新生成变为良好。如果该卷状态不能变为良好，可以尝试恢复该卷。如果不能恢复，则可以尝试分离镜像，之后添加恢复的磁盘

以便重建镜像。

- 如果驱动器状态为联机（错误）、过期数据、失败、失败中或失败冗余，可以使用RECOVER命令刷新镜像集的状态。如果必要就进行恢复，之后对镜像卷进行重新同步。如果一切正常，则驱动器状态应该逐渐从正在重新生成（或再同步）变为良好。如果该卷状态不能变为良好，可以使用BREAK命令终止镜像，之后使用ADD命令在恢复的磁盘或新磁盘上重建镜像。
- 如果某个驱动器状态为不可读，可以使用RESCAN命令重新扫描系统上的驱动器。如果驱动器状态没有改变，则可能需要重新引导计算机。
- 如果无法使得某个驱动器联机，可以分离镜像，指定失效磁盘为待移除磁盘。替换或恢复该磁盘，之后使用ADD命令重建镜像。

真实场景 镜像驱动器的失效将阻止系统进行引导，这主要发生在对系统卷或引导卷进行镜像后主镜像驱动器失效等情况。对这种情况，你需要修改引导配置数据，以便将镜像集中的从驱动器用于系统启动。更多信息可以参考 *Windows Server 2008 Administrator's Pocket Consultant* 一书的第13章。

12.4.3 修复不带奇偶校验信息的 RAID-0 条带集

如前面所讨论的，RAID-0并不能提供容错功能。如果RAID-0中的某个磁盘失效，则整个条带集将不可用。在尝试恢复条带集之前，你应该修复或替换失效的驱动器，之后再重建RAID-0集，并从备份中恢复数据。

12.4.4 重建带奇偶校验信息的 RAID-5 条带集

如果单一驱动器失效，RAID-5可以恢复条带集。如果条带集状态变为失败冗余，则可以获知某个驱动器失效，所采取的纠正措施则依赖于驱动器的具体状态，如下所示。

- 如果驱动器状态为丢失、脱机或良好（有风险），检查该驱动器是否供电，以及物理连接是否正确。之后，启动DiskPart，使用ONLINE命令将该卷联机。如果一切正常，则驱动器状态应该逐渐从正在重新生成变为良好。如果该卷状态不能变为良好，可以尝试恢复该卷。如果不能恢复，你可能需要使用REPAIR命令。
- 如果驱动器状态为联机（错误）、过期数据、失败、失败中或失败冗余，可以使用RECOVER命令刷新RAID-5集的状态。如果必要就进行恢复，之后对其进行重新同步。如果一切正常，则驱动器状态应该逐渐从正在重新生成（或再同步）变为良好。如果驱动器状态不能变为良好，则可能需要使用REPAIR命令。
- 如果某个驱动器状态为不可读，可以使用RESCAN命令重新扫描系统上的驱动器。如果驱动器状态没有改变，则可能需要重新引导计算机。
- 如果某个驱动器无法联机，则可能需要使用REPAIR命令。

你可以使用REPAIR命令修复RAID-5。如果可能，在进行这一操作之前，你应该对数据进行备份，这将确保在发生故障时恢复数据。要解决RAID-5集中的问题，可以遵循如下步骤。

- (1) 在命令提示符中键入**diskpart**，调用DiskPart。
- (2) 列出计算机上的磁盘，以便确认RAID-5集是否确已失效，使用如下命令：

```
DISKPART> list disk
```


(3) 如果必要并且可能，移除并替换失效的驱动器。之后指定新驱动器作为RAID-5集的一部分，使用如下命令：

```
DISKPART> repair disk=n
```

其中，*n*指定了用于替换失效的RAID-5驱动器的动态磁盘。要注意的是，指定的磁盘必须包含足够的空闲空间（等于或大于失效RAID磁盘上已用空间总量）。此外，你可以使用Align参数，以便将新卷上分区或卷起点与最近邻的对齐边界对齐。



Part 4

第四部分

使用命令行管理 Windows 活动目录

本 部 分 内 容

- 第 13 章 核心目录服务管理
- 第 14 章 管理计算机账号与域控制器
- 第 15 章 管理活动目录用户与组

活动目录域服务是Windows网络的一个重要领域。活动目录是一种可扩展的目录服务，提供了一个网络范围的数据库，用于存储账号与资源信息。通过使用活动目录，可以提供一种统一的方式，以便对资源信息进行命名、描述、定位、管理以及安全维护。也就是说，可以使用活动目录方便地操作用户、组、计算机账号，就像操作应用程序、文件、打印机以及其他资源一样简单。可以使用活动目录来管理网络基础设施、执行系统管理任务，并对用户环境进行控制。

只有在包含域控制器的Windows域中，活动目录才是可用的。域控制器是一台Windows服务器，其上安装了活动目录域服务这一角色。活动目录充当了确保安全的中枢，但与安全账号管理者（SAM）不同的是，活动目录同时还充当了不同系统整合点的作用。活动目录将各种管理任务整合到一个基于Windows的管理工具集中，其中对应的目录服务命令行工具将在本章后面分别进行讨论。

注解 要从运行Business版、Enterprise版或Ultimate版Windows Vista上管理活动目录，必须为其安装微软远程服务器管理工具，具体信息可参考1.2.1节。

13.1 从命令行控制活动目录

要有效使用那些用于管理活动目录的命令行工具，需要对活动目录及其结构有一个基本的理解。在设计活动目录时，微软使用域名系统（DNS）作为命名机制。通过DNS，使用一种层次化的结构对网络资源进行组织，这种层次化结构与对网络资源的管理层面相对应。域层次结构，或者称之为域树，是活动目录环境的骨架，形式上与文件系统中使用的目录结构类似，这种层次结构也可以称之为命名空间。每个使用活动目录域的组织都有自己的活动目录层次化结构或命名空间。

13.1.1 理解域、容器与对象

你所创建的第一个活动目录域称为域树的根域，或者称为其下所有子域的父域，根域下的所有域称为子域。比如，假定所创建的根域为cpandl.com。对于其下子域的划分，可以根据地理位置划分或根据功能划分。如果根据地理位置划分，可以设置seattle.cpandl.com、ny.cpandl.com、la.cpandl.com等子域；如果根据功能划分，则可以设置sales.cpandl.com、support.cpandl.com、tech.cpandl.com等子域。这里的关键是，子域名必须从父域名扩展而来。如果某个域名不是从父域名扩展而来，则该域名位于另外的命名空间。比如microsoft.com、msn.com、hotmail.com等域名都不与cpandl.com在同一个命名空间。

如果需要，也可以在命名空间内创建附加的层次。比如，在子域la.cpandl.com内，可以创建sales.la.cpandl.com、tech.la.cpandl.com、support.la.cpandl.com等子域。如果这些还不够，你还可以在命名空间中创建更多的子域层次。要记住的是，活动目录会对域树内的关系进行管理，并为域创建适当的信任关系。

在活动目录中，信任关系是双向的、传递的。简单地创建一个子域，比如tech.la.cpandl.com，就自动地在tech.la.cpandl.com与la.cpandl.com之间建立了双向的信任关系，并且这种信任关系还可以在域树内传递。也就是说，由于la.cpandl.com信任cpandl.com，tech.la.cpandl.com与cpandl.com之间自动地建立了双向的信任关系，同时也自动地与该命名空间内的所有其他子域建立了信任关系。

活动目录使用对象来表示网络资源，比如用户、组、计算机等。还使用一种称之为容器的专门对象来对网络资源进行组织，进行组织的依据可以是地理位置、商业、功能需求等。典型情况下，容器用于将具有相似属性的对象组合在一起。比如，你可能需要为所有工程师赋予一组特定的许可权限，为此，将所有这些用户放置到一个容器中是一个简单易行的方法。

每个容器代表一组对象，每个单独的资源由一个独一无二的活动目录对象进行表示。最常见的活动目录容器类型为组织单元，或称之为OU。放置到同一个OU中的对象只能来自相关的域，比如，与tech.la.cpandl.com相关联的多个OU只能包含来自该域的对象。也就是说，你不能向这些容器中添加来自support.la.cpandl.com、la.cpandl.com或tech.ny.cpandl.com等域的对象。

每个活动目录对象类，比如容器、用户、组、打印机等，都分配了一组属性，用于描述单独的资源。比如，用户对象包含了用于描述用户账号的属性，包括联系信息、许可权限、特权等。也就是说，用户对象属性与首名、尾名、显示名、电话号码、电子邮件地址、口令等相关。

由于活动目录内的每个对象实质上是数据库内的一条记录，因此可以对属性集进行扩展，来满足不同组织的需要。也就是说，可以为其添加自定义属性，来更好地描述对象。比如，你可以添加一个属性，用于描述雇员的身份代码。

13.1.2 理解活动目录中的逻辑结构与物理结构

到这里为止，对活动目录结构的讨论主要是目录数据逻辑上的组织结构，包括域、子域、OU等。根据商业或功能需求，你可以使用这些结构对活动目录进行组织。你也可以按照地理位置进行组织，比如seattle.cpandl.com、ny.cpandl.com、la.cpandl.com等子域。

然而，对域、子域、OU等的划分和定义，仍然没有将其映射到真实的物理世界——即便根据地理位置进行分解，但仍然是目录内的一些用于存储相关信息的逻辑位置，而这些逻辑结构都应该有真实对应的物理结构。

在物理世界中，域、子域、OU等逻辑结构都可能跨越几个物理区域。不管是某个建筑物的不同层、不同的建筑物、或者不同的城市，都统一看做不同的物理位置。要想使得活动目录理解这些不同的位置，必须定义子网与站点。子网是一些计算机和网络设备的集合，这些计算机和网络设备的IP地址位于某个特定的区域，并有一个子网掩码。站点则是一个或多个子网的组合，映射了网络的物理结构。由于站点映射独立于逻辑域结构，因此，网络的物理结构与其逻辑域结构没有必然的联系。

你可以在单一域内创建多个站点，也可以创建一个站点为几个域所用。比如，如果根据地理位置对子域进行分组，你可以创建seattle.cpandl.com、ny.cpandl.com、la.cpandl.com等几个子域，分别对应Seattle-Site、NY-Site、LA-Site等站点；如果你所在组织只有一个办公室，并根据功能进行子域划分，就可以创建sales.la.cpandl.com、tech.la.cpandl.com、support.la.cpandl.com等几个子域，并建立一个站点

Main-Site。

13.1.3 理解区分名

活动目录中的每个对象都有相关联的区分名，即DN。根据对象的通常名及其在相应命名空间中的位置，DN唯一标识了该对象。对象的通常名是在创建对象时赋予该对象的，以明文的英文形式呈现。通过CN=*Name*，可以标识对象的通常名，其中，*Name*就是对象的通常名，比如：

CN=William Stanek

对象的通常名也是该对象的相对DN (RDN)。也就是说，对象全名的这一部分代表了该对象在活动目录中的位置。对象的位置是由包含该对象的容器对象与域名决定的。你可以使用OU= 来标识OU容器，使用DC= 来标识域组件。域树内的每一层次都可以分解为单独的域组件，参考如下实例：

OU=Engineering, DC=ny, DC=tech, DC=cpandl, DC=com

上面的实例中，为ny.tech.cpandl.com域内的Engineering OU指定了DN，每个名组件使用逗号分隔开，名组件从域树的最底层到最顶层。也就是说，从包含待操作具体对象的OU，到子域、到父域，直至最后到根域。

DN之所以如此重要，是因为DN指定了对象的确切位置。活动目录使用DN来搜索、取回、管理数据库内的对象，获知某对象的DN后，就可以执行这些任务。

所有对象都有相关联的容器与域组件。尽管典型情况下，用户、计算机、组以及其他类型对象都是OU，但也并非总是如此。因为活动目录还包含了几个默认的容器，其中可以存储对象。通过CN= 语句，这些默认的容器使用其通用名进行标识，包括下面4个。

- Builtin。用于内置安全组的容器。
- Computers。用于域内成员服务器与工作站的默认容器。
- ForeignSecurityPrincipals。用于来自可信外部域的对象容器。
- Users。用户的默认容器。

注解 域控制器容器创建为一个OU。也就是说，你可以使用OU=Domain Controller作为名标识符。

以这些信息为基础，可以对这些容器内的对象进行标识和识别。比如，如果需要对tech.cpandl.com域内的Users容器中的对象进行标识，可以使用如下语句：

CN=Users, DC=tech, DC=cpandl, DC=com

如果该对象是William Stanek的用户账号，则完整的DN为：

CN=William Stanek, CN=Users, DC=tech, DC=cpandl, DC=com

如果用户账号后来移动到Engineering OU，其DN为：

CN=William Stanek, OU=Engineering, DC=tech, DC=cpandl, DC=com

13.1.4 使用活动目录命令行工具

获知了活动目录的基本结构、并可以识别待操作对象的DN后，就可以从命令行控制活动目录。从命令行对活动目录进行控制和操作的关键优势在于其更多的灵活性。实际上，你可以在命令行中

便地执行很多任务，而在图形用户界面中，执行这些任务或者非常复杂，或者无法执行。比如，你可以搜索那些处于不活跃状态超过一个星期的计算机账号，之后禁用它。或者你可以在单一命令中同时修改多个用户账号的属性。

在操作Windows域时，Windows Server 2008包含了一组命令行工具，可以对活动目录进行管理。对于Business版、Enterprise版或Ultimate版的Windows Vista，要使用活动目录工具，需要先安装Windows Server 2008管理工具包，这些工具包括下面列举的。

- DSADD。向活动目录中添加对象。
- DSGET。显示活动目录中注册对象的属性。
- DSMOD。修改活动目录中对象的属性。
- DSMOVE。在单一域内将单一对象移动到新位置，或者对其重命名而不移动。
- DSQUERY。使用特定搜索标准寻找活动目录中的对象。
- DSRM。从活动目录中移除对象。

上面每个命令行工具都可以用于操作活动目录中一组特定的对象，表13-1总结了这些工具可以操作的对象。

表13-1 活动目录命令行工具及其可操作的对象

对 象	Dsquery	Dsget	Dsadd	Dsmmod
Computer	是	是	是	是
Contact	是	是	是	是
Group	是	是	是	是
Partition	是	是	否	是
Quota	是	是	是	是
Server	是	是	否	是
Site	是	是	否	否
Subnet	是	是	否	否
User	是	是	是	是
OU	是	是	是	是

大多数情况下，对活动对象的操作需要使用一些特定于待操作对象类型的参数，用于访问这些参数的子命令名则与待操作对象名相同。比如，如果需要向域中添加一台计算机，则可以使用DSADD COMPUTER命令以及相关的参数。如果需要向域中添加一个用户账号，则可以使用DSADD USER命令以及相关的参数。

注解 表13-1中没有列出DSMOVE命令与DSRM命令，因为这两条命令可以操作活动目录中的任意对象，你可以根据对象的DN来移动或删除对象。此外，通过使用*作为DSQUERY操作的对象名，可以在目录中查找任意符合查询标准的对象。

13.2 使用 DSQUERY 命令进行目录查询

你可以使用DSQUERY命令在活动目录中进行查询，以便搜索符合特定标准的对象。比如，你可以搜索所有以D引导的计算机账号，或者所有处于禁用状态的计算机账号，DSQUERY会返回符合查

询标准的对象列表。

13.2.1 DSQUERY 子命令及语法

你可以使用如下一些子命令及其相应语法来查询活动目录。

- **DSQUERY COMPUTER**。搜索符合指定标准的计算机账号。

```
dsquery computer [{StartNode | forestroot | domainroot}]
[-o {dn | rdn | samid}] [-scope {subtree | onelevel | base}]
[-name Name] [-desc Description] [-samid SAMName]
[-inactive NumberOfWeeks] [-stalepwd NumberOfDays] [-disabled]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-q] [-r]
[-gc] [-limit NumberOfObjects] [{-uc | -uco | -uci}]
```

- **DSQUERY CONTACT**。搜索符合指定标准的联系人。

```
dsquery contact [{StartNode | forestroot | domainroot}]
[-o {dn | rdn}] [-scope {subtree | onelevel | base}] [-name Name]
[-desc Description] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-q] [-r] [-gc] [-limit NumberOfObjects]
[{-uc | -uco | -uci}]
```

- **DSQUERY GROUP**。搜索符合指定标准的组。

```
dsquery group [{StartNode | forestroot | domainroot}]
[-o {dn | rdn | samid}] [-scope {subtree | onelevel | base}]
[-name Name] [-desc Description] [-samid SAMName]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-q] [-r]
[-gc] [-limit NumberOfObjects] [{-uc | -uco | -uci}]
```

- **DSQUERY OU**。搜索符合指定标准的OU。

```
dsquery ou [{StartNode | forestroot | domainroot}] [-o {dn | rdn}]
[-scope {subtree | onelevel | base}] [-name Name] [-desc Description]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-q]
[-r] [-gc] [-limit NumberOfObjects] [{-uc | -uco | -uci}]
```

- **DSQUERY PARTITION**。搜索符合指定标准的活动目录分区。

```
dsquery partition [-o {dn | rdn}] [-part Filter]
[-desc Description] [{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-q] [-r] [-limit NumberOfObjects]
[{-uc | -uco | -uci}]
```

- **DSQUERY QUOTA**。搜索符合指定标准的对象配额。

```
dsquery quota {domainroot | ObjectDN} [-o {dn | rdn}] [-acct Name]
[-qlimit Filter] [-desc Description] [{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-q] [-r] [-limit NumberOfObjects]
[{-uc | -uco | -uci}]
```

- **DSQUERY SERVER**。搜索符合指定标准的域控制器。

```
dsquery server [-o {dn | rdn}] [-forest] [-domain DomainName]
[-site SiteName] [-name Name] [-desc Description]
[-hasfsmo {schema | name | infr | pdc | rid}] [-isgc]
[-isreadonly] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-q] [-r] [-gc] [-limit NumberOfObjects]
```

```
[{-uc | -uco | -uci}]
```

- **DSQUERY SITE**。搜索符合指定标准的活动目录站点。

```
dsquery site [-o {dn | rdn}] [-name Name] [-desc Description]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}]
[-q] [-r] [-gc] [-limit NumberOfObjects] [{-uc | -uco | -uci}]
```

- **DSQUERY SUBNET**。搜索符合指定标准的子网对象。

```
dsquery subnet [-o {dn | rdn}] [-name Name] [-desc Description]
[-loc Location] [-site SiteName] [{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-q] [-r] [-gc]
[-limit NumberOfObjects] [{-uc | -uco | -uci}]
```

- **DSQUERY USER**。搜索符合指定标准的用户账号。

```
dsquery user [{StartNode | forestroot | domainroot}]
[-o {dn | rdn | upn | samid}] [-scope {subtree | onelevel | base}]
[-name Name] [-namep namephonetic] [-desc Description] [-upn UPN]
[-samid SAMName] [-inactive NumberOfWeeks] [-stalepwd NumberOfDays]
[-disabled] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-q] [-r] [-gc] [-limit NumberOfObjects]
[{-uc | -uco | -uci}]
```

- **DSQUERY ***。搜索符合指定标准的任意活动目录对象。

```
dsquery * [{StartNode | forestroot | domainroot}]
[-scope {subtree | onelevel | base}] [-filter LDAPFilter]
[-attr {AttributeList | *}] [-attrsonly] [-l]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}]
[-q] [-r] [-gc] [-limit NumberOfObjects] [{-uc | -uco | -uci}]
```

一瞥之下，会感觉这些命令的语法格式太过复杂了。但实际上并非如此，大多数DSQUERY子命令都使用一种标准的语法格式，只是在针对操作的具体对象时加上一些特定的扩展。学习DSQUERY子命令的最佳途径是去实践它，下面将通过实例来讲解这些子命令。

13.2.2 使用名称、描述、SAM 账号名进行搜索

撇开使用的其他参数不谈，DSQUERY的搜索语法应该包含名称、描述、或待搜索的SAM账号名。使用-Name参数时，DSQUERY将搜索指定类型的对象，这些对象名与给定值匹配。你可以使用*作为通配符，用于对部分名进行搜索与匹配。比如，使用参数-Name Will*将会匹配William Stanek。对名进行简单搜索时，可以使用类似于如下的语法格式：

```
dsquery user -name Will*
```

这一命令的输出信息中包含了与指定搜索标准匹配的任意用户账号的DN，比如：

```
"CN=william R. Stanek, CN=Users, DC=cpandl, DC=com"
```

上面给出的就是进行简单搜索时的操作方法，只使用了一个参数，就获得了想要的结果。

注解 对于用户对象，-Name参数搜索的是用户属性对话框中的显示名。在上面的实例中，显示名为William R. Stanek。对于其他类型的对象，-Name参数指定的是该对象属性对话框的“常规”选项卡中的名称字段。

星号(*)可以出现在搜索标准中的任意位置。如果已知某用户的尾名,而不知该用户的首名,则可以根据尾名进行搜索,比如:

```
dequery user - name *Stanek
```

你也可以用部分首名、尾名进行搜索,比如:

```
dequery user - name W*Stanek
```

使用**-desc**参数时,搜索的是特定类型的对象,这些对象的描述信息与该参数指定的值匹配。使用星号作为通配符,可以使用部分描述信息进行匹配,比如,**-desc Eng***可以用于匹配Engineering Workstation。参考如下实例:

```
dsquery computer -desc Server*
```

这一命令的输出信息中包含了与指定搜索标准匹配的任意计算机账号的DN,比如:

```
"CN=CORPSVR02,OU=Domain Controllers,DC=cpandl,DC=com"
```

注解 **-Desc**参数搜索的是对象属性对话框中列出的描述字段,前面的实例中,该计算机账号的描述信息以Server引导。

使用**-Samid**参数时,搜索的是特定类型的对象,这些对象的SAM账号名与该参数指定的值匹配。使用星号作为通配符,可以使用部分SAM账号名进行匹配,比如,**-samid wr***可以用于匹配wrstanek。

注解 在用户的属性对话框中,SAM账号名是在“账号”选项卡中列出的用户登录名,对计算机与组,SAM账号名与相关的账号名相同。

13.2.3 设定搜索的登录域与 Run As 许可权限

默认情况下,使用DSQUERY时,将连接到登录域内的某个域控制器。通过使用**-S**参数,并在其后跟随服务器的DNS名,也可以连接到任意域内的特定域控制器,比如:

```
-s corpd01, cpandl.com
```

这一实例中,连接到cpandl.com内的域控制器corpd01。

注解 技术上讲,不需要指定服务器的完全限定域名(DNS名),可以只是使用服务器名。然而,这种做法会降低搜索速度,因为活动目录必须先进行DNS查询来获取完整名,之后才能进行具体的查询。

你也可以连接到某个域内任意可用的域控制器,而不是局限于某个特定的域控制器。这是通过使用**-D**参数实现的,其后跟随该域的DNS名,比如:

```
-d tech.cpandl.com
```

通过这一语法,你可以连接到tech.cpandl.com域内的任意可用的域控制器。要注意的是,不能同时使用**-S**参数与**-D**参数。也就是说,你或者可以连接到某个特定的域控制器,或者可以连接到给定域内任意可用的域控制器。

与很多其他类型命令类似，必要的时候，你可以使用用户名与口令进行身份认证，使用如下的语法格式：

```
-u [Domain\]User [-p Password]
```

其中，*Domain*为可选的域名，用户账号就存在于该域内，*User*为用户账号名（要使用的就是该用户账号的许可权限），*Password*为该用户账号的口令（可选）。如果没有指定域，则系统会假定当前域作为默认的域名；如果没有指定口令，则会弹出提示信息要求输入口令。

要了解如何结合使用这些参数，参考如下实例。

使用cpandl logon域内的Wrstanek用户账号，连接到tech.cpandl.com域内的corpsvr02域控制器，搜索显示名以Stanek结尾的用户账号：

```
dsquery user -name *Stanek -s corpsvr02.tech.cpandl.com -u  
cpandl\wrstanek
```

使用cpandl logon域内的Wrstanek用户账号，连接到tech.cpandl.com域内任意可用的域控制器，搜索显示名以Will开始的用户账号：

```
dsquery user -name Will* -d tech.cpandl.com -u cpandl\wrstanek
```

13.2.4 设定开始节点、搜索范围与对象限制

在命令语法中，开始节点是使用{*StartNode* | **forestroot** | **domainroot**}定义的，也可能包含*ObjectDN*。开始节点的主要作用是规定搜索的起点。你可以指定森林根（键入**forestroot**）、域根（键入**domainroot**）、节点DN（*StartNode*）等信息，比如“**CN=Users,DC=cpandl,DC=com**”。如果指定的是**forestroot**，则搜索会使用全局编目。但默认值为**domainroot**，即搜索是从当前用户账号登录域的顶级容器开始的。有些子命令还可以传送待操作对象的实际DN（*ObjectDN*），比如“**CN=William Stanek, CN=Users,DC=cpandl,DC=com**”

注解 你可能已经注意到，我使用双引号封装了对象DN。这种做法是有益的，因为如果DN中包含空格（如上面第2个对象DN），就必须使用双引号。

如果需要进行彻底的搜索，你可能需要指定节点的DN，如果希望返回完整的对象集，这种做法的价值就会体现出来。比如，通过指定开始节点（而不需要指定-Name、-Desc、-Samid等参数），就可以返回某特定容器内特定类型的所有对象列表。

要了解如何使用开始节点，可以参考如下实例。

返回某个域内的所有计算机账号列表：

```
dsquery computer "DC=cpandl,DC=com"
```

返回Computer容器内所有计算机账号列表：

```
dsquery computer "CN=Computers,DC=cpandl,DC=com"
```

返回 Domain Controller OU内的所有计算机列表：

```
dsquery computer "OU=Domain Controllers,DC=cpandl,DC=com"
```


返回某个域内的所有用户列表:

```
dsquery user "DC=cpan1,DC=com"
```

返回Users容器内所有用户列表:

```
dsquery user "CN=Users,DC=cpan1,DC=com"
```

返回 Tech OU内的所有用户列表:

```
dsquery user "OU=Tech,DC=cpan1,DC=com"
```

除了指定开始节点外,你也可以指定搜索范围。在命令语法中,搜索范围是使用{-scope **subtree** | **onelevel** | **base**}指定的。默认情况下,使用的搜索范围是**subtree**,表示搜索范围以开始节点为根节点的**subtree**。如果使用**subtree**,对**domainroot**而言,搜索范围是整个域;对**forestroot**而言,搜索范围是整个森林;对特定的容器而言,搜索范围是指定的容器与任意子容器。比如,如果开始节点设置为“OU=Tech,DC=cpan1,DC=com”,则活动目录将搜索Tech OU以及其内的任意OU。

你可以使用**onelevel**来为指定的开始节点及其直接子节点设置范围。比如,对**domainroot**而言,这种做法意味着包含该域及其顶层容器与OU。然而,如果某些OU包含了附加的(子)OU,就不会被列入搜索范围。

如果使用**base**设置范围,则仅对开始节点代表的单一对象进行搜索。比如,只搜索指定的OU,不搜索其子OU。

注解 在**forestroot**被设置为开始节点时, **subtree**是唯一有效的搜索范围。

要了解如何使用搜索范围,参考如下实例。

搜索Tech OU及其下子OU内的计算机账号:

```
dsquery computer "OU=Tech,DC=cpan1,DC=com"
```

注解 默认的搜索范围是**subtree**,即**-scope subtree**是自动隐含的。

只搜索Tech OU内的计算机账号:

```
dsquery computer "OU=Tech,DC=cpan1,DC=com" -scope base
```

搜索Tech OU及其下直系子OU内的计算机账号:

```
dsquery computer "OU=Tech,DC=cpan1,DC=com" -scope onellevel
```

另一个可选的参数是**-Limit**,用于设置搜索结果中返回对象数的最大值。默认情况下,如果没有指定这一数值,则前100个结果会显示出来。如果需要设定不同的范围,可以在该参数后跟随要返回的对象数。比如,如果只需要显示前10个结果,就可以键入**-limit 10**。要去除对象限制,显示所有匹配结果,可以使用值0,比如, **-limit 0**。

提示 在可能包含数千个对象的大型组织中，不应该去除对象限制，而应该设定返回对象的限定值，或者使用默认的设置。这将确保所做的查询不会给正在操作的域控制器造成过大的负载。

13.2.5 设定名的输出格式

通过DSQUERY，可以设置返回名值的输出格式以及单个字符的格式。在命令语法中，名的输出格式是使用-o定义的，其后跟随下列元素中的一个：{dn|rdn|upn|samid}。默认情况下，输出格式是DN（使用-o dn指定），比如“CN=William R.Stanek,CN=Users,DC=cpandl,DC=com”。你也可以将输出格式指定为相对DN（使用-o rdn指定）、用户主要名（使用-o upn指定）或SAM账号名（使用-o samid指定）。

RDN是对象的常用名，取自DN的底层名部分。对用户而言，RDN与相关联的属性对话框中的显示名是相同的。对其他类型对象，RDN是该对象属性对话框的“常规”选项卡中的名称字段。下面给出了几个RDN示例：

- ❑ “William R. Stanek”;
- ❑ “CORPSVR01”;
- ❑ “Administrators”。

UPN只适用于用户账号。在活动目录中，存在一个与UPN对应的实际字段，用于登录与认证。在用户的属性对话框中，你可以在“账号”选项卡中发现用户的登录名与登录域。比如，wrstanek@cpandl.com就是一个UPN实例。其中，wrstanek是登录名，@cpandl.com是登录域信息。

SAM账号名适用于用户、计算机、组。在活动目录中，也存在与SAM账号名对应的实际字段，你可以在属性对话框中浏览并找到。对用户而言，SAM账号名就是Windows 2000以前系统的账号名，该值在相关属性对话框的“账号”选项卡中指定；对组而言，SAM账号名与“常规”选项卡中的名称字段是相同的；对计算机，SAM账号名与“常规”选项卡中的名称字段是相同的，但其后带了一个美元符号（\$）作为后缀。

注解 美元符号（\$）是实际的计算机账号名的一部分，但通常是隐含的，使用时也不需引用。活动目录使用\$的目的是使得同名的用户账号与计算机账号可以同时存在，但又不至于混淆造成错误。比如，用户JAMESW可以有一个名为JAMESW的计算机，这种做法在Windows 2000之前的操作系统中是无法做到的。

要了解关于名格式的更多信息，参考如下一些实例。

返回符合搜索标准的计算机的RDN：

```
dsquery computer -name corp* -o rdn
```

返回符合搜索标准的用户的SAM账号名：

```
dsquery user -name Wi* -o samid
```

返回符合搜索标准的用户的UPN：

```
dsquery user "OU=Tech,DC=cpandl,DC=com" -o upn
```

返回符合搜索标准的用户的DN:

```
dsquery user "CN=Users,DC=cpand1,DC=com"
```

注解 默认的格式是DN, 也就是说, **-o dn**是自动隐含的。

13.2.6 结合使用 DSQUERY 与其他活动目录命令行工具

由于DSQUERY返回的是匹配对象的DN, 这些结果是有用的, 可以通过管道作为其他活动目录命令行工具的输入信息。下面的实例中, 搜索所有名称以*Willia*引导的用户账号:

```
dsquery user -name Willia*
```

这一命令的输出信息包含了任意符合搜索标准的账号的DN, 比如:

```
"CN=William R. Stanek,CN=Users,DC=cpand1,DC=com"
```

你可以将上述输出信息通过管道提供给DSGET USER命令, 用来显示一个组列表, 这些组成员中包含了这一用户:

```
dsquery user -name Willia* | dsget user -memberof -expand
```

这一命令的输出信息将根据DN展示组成员, 如下:

```
"CN=Domain Admins,CN=Users,DC=cpand1,DC=com"
"CN=Enterprise Admins,CN=Users,DC=cpand1,DC=com"
"CN=Administrators,CN=Builtin,DC=cpand1,DC=com"
"CN=Domain Users,CN=Users,DC=cpand1,DC=com"
"CN=Users,CN=Builtin,DC=cpand1,DC=com"
```

13.3 搜索问题用户与计算机账号

DSQUERY USER与DSQUERY COMPUTER包含了一些扩展的语法格式, 用于搜索那些存在问题的账号。比如, 你可以使用**-Disabled**参数寻找那些已被禁用的账号。如果需要在整个域内搜索禁用的用户账号, 可以使用命令**dsquery user -disabled**。这一命令的输出信息中包含了所有已禁用的计算机账号, 并分别列出其DN, 比如:

```
"CN=Guest,CN=Users,DC=cpand1,DC=com"
"CN=SUPPORT_456945a0,CN=Users,DC=cpand1,DC=com"
"CN=krbtgt,CN=Users,DC=cpand1,DC=com"
```

另一个很有用的命令选项是**-Stalepwd**, 用于搜索那些已经超过指定最多天数而未更改口令的账号。比如, 要搜索那些至少15天没有更改口令的用户账号, 可以使用命令**dsquery user -stalepwd 15**。其输出信息中包含了符合这一标准的用户, 并分别列出其DN, 比如:

```
"CN=Administrator,CN=Users,DC=cpand1,DC=com"
"CN=Guest,CN=Users,DC=cpand1,DC=com"
"CN=SUPPORT_456945a0,CN=Users,DC=cpand1,DC=com"
"CN=krbtgt,CN=Users,DC=cpand1,DC=com"
"CN=William R. Stanek,CN=Users,DC=cpand1,DC=com"
"CN=Howard Smith,CN=Users,DC=cpand1,DC=com"
```

真实场景 你可以设置口令策略，并在其中要求用户定期更改口令，*Windows Server 2008 Administrator's Pocket Consultant* (Microsoft Press, 2008)一书的第10章讨论了这一主题。只有在用户登录域后，这些口令策略才是适用的。如果某用户休假，或者不再可用，则可能会超过这一更改口令的最后期限（但通常在该用户下次登录时，将不得不修改口令）。大多数禁用的账号也将在口令过期账号列表中呈现。

最后，你可能也需要搜索那些处于不活跃状态已经超过指定的最少几星期的计算机或用户账号。不活跃的账号是指在指定的时间间隔内没有登录域的账号。比如，如果需要搜索那些至少两个星期没有登录域的用户账号，可以使用 **dsquery user -inactive 2** 命令。

通常，用户很长时间没有登录域是因为该用户已不在该办公室，比如去休假、生病或者在家工作。对计算机账号，不活跃状态意味着该计算机已关机或断网。比如，某用户休假，随身携带了笔记本，但没有远程连接到办公室，则相关计算机账号在该段时间内将处于不活跃状态。

13.4 对象的重命名与移动

在域内，对象的重命名与移动是使用DSMOVE命令实现的。之所以会使用同一命令完成这两个任务，是因为在对对象进行重命名时，实际上同时也将该对象从当前DN移动到新DN。要记住的是，DN包含两部分：通常名、RDN和位置。

DSMOVE命令的语法格式如下：

```
dsmove ObjectDN [-newname NewName] [-newparent ParentDN]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-q]
[{-uc | -uco | -uci}]
```

要对用户、计算机、组或其他活动目录对象重命名，必须指定该对象的DN，并使用-Newname参数指定新的相对名。比如，如果需要将用户对象William Stanek重命名为William R. Stanek，可以使用命令 **dsmove "CN=William Stanek, OU=Tech, DC=cpandl, DC=Com" -newname "William R.Stanek"**

要在某个域内移动用户、计算机、组或其他活动目录对象，必须指定该对象的当前DN，并使用-Newparent参数指定新位置或该对象的父DN。比如，如果需要将一个用户账号从Tech OU移动到Engineering OU，该对象的DN为 **"CN=William Stanek, OU=Tech, DC=cpandl, DC=Com"**，新位置的DN为 **"OU=Engineering, DC=cpandl, DC=com"**，则相关的命令语法格式如下：

```
dsmove "CN=William Stanek, OU=Tech, DC=cpandl, DC=com"
-newparent OU=Engineering, DC=cpandl, DC=com
```

如果需要在移动某个对象的同时对其重命名，只需要使用-Newname参数为该对象赋予一个新名，参考如下实例：

```
dsmove "CN=William Stanek, OU=Tech, DC=cpandl, DC=com"
-newparent OU=Engineering, DC=cpandl, DC=com -newname "William R.Stanek"
```

上面的命令中，将用户账号William Stanek移动到Engineering OU，并将其重命名为William R. Stanek。

在上面这些实例中，都是假定先通过DSQUERY获取了对象DN。实际上，可以将DSQUERY的输出信息通过管道传递给DSMOVE命令，如下所示：

```
dsquery user -name "William Stanek" | dsmove -newname "William R. Stanek"
```

其中,对象的DN “CN=William Stanek, OU=Tech,DC=cpandl,DC=Com” 是通过DSQUERY USER命令获取的,并用作DSMOVE的输入信息,最终实现对用户对象的重命名。

提示 需要在不同域间移动对象么? 如果需要,可以使用活动目录迁移工具,在微软下载站点(<http://downloads.microsoft.com>) 可以获取该工具,要确保获取与Windows Server 2008兼容的版本。

13.5 从活动目录中移除对象

如果某对象不再需要存在于活动目录中,可以使用DSRM命令永久性地删除该目录,DSRM的语法格式如下:

```
dsrm ObjectDN ... [-subtree [-exclude]] [-noprompt] [{-s Server |
-d Domain}] [-u UserName] [-p {Password | *}] [-c] [-q] [{-uc | -uco |
-uci }]
```

警告 使用DSRM命令之前,应该先在独立的测试域内对其进行实验。这一命令功能强大,可以删除提交给他的任意对象,包括对象容器。

使用DSRM的最佳方法是提交一个待删除的特定对象。在下面的实例中,从cpandl.com域内的Eng OU中删除了计算机账号engcomp18:

```
dsrm "CN=engcomp18,OU=eng,DC=cpandl,DC=com"
```

默认情况下,DSRM会弹出提示信息,询问是否确定删除:

确认删除CN=engcomp18,OU=Eng,DC=cpandl,DC=com (是/否)?

你可以使用-noprompt参数禁用这种确认提示,如下所示:

```
dsrm "CN=engcomp18,OU=Eng,DC=cpandl,DC=com" -noprompt
```

然而,只有在确信DSRM删除的确实是需要删除的对象时,才可以这样做。

DSRM可以用于删除容器或OU内的对象,也可以用于删除容器或OU本身。如果容器或OU是空的,可以根据其DN删除,比如:

```
dsrm "OU=Eng,DC=cpandl,DC=com"
```

如果容器或OU非空,则不能按照上面的方法删除,DSRM会报告错误信息:

“失败:操作无法进行,因为存在于对象。该操作只能在叶子对象上进行。”

要删除容器及其包含的所有对象,可以使用-Subtree参数,参考如下实例:

```
dsrm "OU=Eng,DC=cpandl,DC=com" -subtree
```

其中,-Subtree参数的作用是从Eng OU中删除所有对象(而不管其类型)以及该容器本身。要删除所有对象,但不删除容器本身,可以结合使用-Subtree参数与-Exclude参数,参考如下实例:

```
dsrm "OU=Eng,DC=cpandl,DC=com" -subtree -exclude
```

其中,-Subtree参数的作用是从Eng OU中删除所有对象(而不管其类型),-Exclude参数的作用是将Eng OU排除在删除范围之外。

本章将讨论对域计算机账号（用于对网络及其自身资源进行访问控制）的管理。与用户账号一样，域计算机账号也包含了一些可以管理的属性，包括名、组成员关系等。可以将计算机账号添加到活动目录内的任意容器与OU，但可以使用的最佳容器是计算机、域控制器以及所创建的OU。用于操作计算机账号的标准微软工具是活动目录用户和计算机。在命令行中，有很多可用的命令，每个命令有其特定的用途。不管你使用的是Windows Vista系统还是Windows Server 2008系统，均可使用本章讲述的技术管理计算机账号与域控制器。

14.1 从命令行管理计算机账号概览

有两组工具可用于管理域计算机账号。一组可用于任意类型的计算机账号，包括工作站、成员服务器以及域控制器；另一组则只能用于域控制器，主要作用是对其附加功能与属性进行管理。

除前面讨论的DSQUERY COMPUTER命令之外，通常的计算机账号管理命令还包括下面列举的。

□ **DSADD COMPUTER**。在活动目录中创建一个计算机账号。

```
dsadd computer ComputerDN [-samid SAMName] [-desc Description]
[-loc Location] [-memberof GroupDN ...] [{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-q] [{-uc | -uco | -uci}]
```

□ **DSGET COMPUTER**。显示计算机账号的属性，有两种命令语法。

用于查看多台计算机属性的语法如下：

```
dsget computer ComputerDN ... [-dn] [-samid] [-sid] [-desc] [-loc]
[-disabled] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
[-part PartitionDN] [-qlimit] [-qused]
```

用于查看单一计算机成员信息的语法如下：

```
dsget computer ComputerDN [-memberof [-expand]]
[{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
```

□ **DSMOD COMPUTER**。对活动目录中一个或多个计算机账号的属性进行修改，其语法格式如下：

```
dsmod computer ComputerDN ... [-desc Description] [-loc Location]
```

```
[ -disabled {yes | no} ] [ -reset ] [ {-s Server | -d Domain} ]
[ -u UserName ] [ -p {Password | *} ] [ -c ] [ -q ] [ {-uc | -uco | -uci} ]
```

提示 对任意计算机与服务器命令，都可以使用来自DSQUERY的输入信息，来指定待操作的对象。如果需要为待操作的每个对象键入区分名（DN），可以用空格对多个DN进行分隔。

除前面讨论的DSQUERY SERVER命令之外，可用于对域控制器附加功能进行管理的工具还包括：

□ **DSGET SERVER**。显示域控制器的多种属性，有3种语法。

用于显示域控制器通常属性的语法如下：

```
dsget server ServerDN ... [ -dn ] [ -desc ] [ -dnsname ] [ -site ]
[ -isgc ] [ {-s Server | -d Domain} ] [ -u UserName ]
[ -p {Password | *} ] [ -c ] [ -q ] [ -l ] [ {-uc | -uco | -uci} ]
```

用于显示安全主体列表，这些安全主体拥有指定域控制器内最大数量的目录对象，其语法为：

```
dsget server ServerDN ... [ {-s Server | -d Domain} ] [ -u UserName ]
[ -p {Password | *} ] [ -c ] [ -q ] [ -l ] [ {-uc | -uco | -uci} ]
[ -topobjowner NumbertoDisplay ]
```

用于指定服务器上目录分区的DN，语法为：

```
dsget server ServerDN ... [ {-s Server | -d Domain} ] [ -u UserName ]
[ -p {Password | *} ] [ -c ] [ -q ] [ -l ] [ {-uc | -uco | -uci} ] [ -part ]
```

□ **DSMOD SERVER**。用于修改域控制器的属性，语法如下：

```
dsmod server ServerDN ... [ -desc Description ] [ -isgc {yes | no} ]
[ {-s Server | -d Domain} ] [ -u UserName ] [ -p {Password | *} ] [ -c ]
[ -q ] [ {-uc | -uco | -uci} ]
```

注解 另一个用于操作域控制器与活动目录的有用命令是NTDSUtil。NTDSUtil是一个文本模式的可调用的命令解释器，可以在其中使用单独的命令提示符与子命令来管理目录服务。要调用NTDSUtil，可以在命令窗口中键入ntdsutil，之后按Enter键。

14.2 在活动目录域内创建计算机账号

通过DSADD COMPUTER命令，可以为工作站或服务器创建计算机账号，并将其添加到域中。一般要预先创建计算机账号，以便在该计算机需要加入到域中时直接使用。要创建计算机账号，必须具备适当的许可权限。大多数用户可以在其登录域内创建计算机账号，但组策略与其他许可权限也可能改变这种情况。

14.2.1 创建计算机账号

创建计算机账号时，唯一需要的信息就是该账号的DN。第13章已经讲过，DN指定了活动目录中

某对象的完整名，并包含了该对象位置的访问路径。有鉴于此，在为计算机账号提供DN时，你需要指定计算机账号名以及该账号所在容器。参考如下实例：

```
dsadd computer "CN=CORPSEVER05,OU=Domain Controllers,DC=cpandl,
DC=com"
```

提示 DN指定了在域层次结构内，计算机账号应该创建的位置。你可以在森林中的任意域内创建计算机账号，前提是具备适当的访问许可权限。有些情况下，你需要直接登录待操作域的域控制器。通过-S Server参数，可以连接到森林中任意域内的特定域控制器；通过-D Domain参数，可以连接到指定域内任意可用的域控制器。

上面的命令中，在活动目录内的Domain Controllers容器内创建了计算机账号CORPSEVER05。如果创建成功，则DSADD COMPUTER会报告如下信息：

```
dsadd成功:CN=CORPSEVER05,OU=Domain Controllers,DC=cpandl,
DC=com
```

使用-U *UserName*与-P *Password*可以设置以哪种许可权限运行。

账号创建并不能保证总是成功，导致账号创建失败最常见的原因是指定了错误的DN。比如，如果使用如下命令：

```
dsadd computer "CN=CORPSEVER05,CN=Domain Controllers,DC=cpandl,
DC=com"
```

则DSADD COMPUTER会报告如下信息：

```
dsadd 失败:CN=CORPSEVER05,CN=Domain Controllers,DC=cpandl,
DC=com:无法找到目录对象
```

之所以产生这一错误，是因为Domain Controllers创建为一个组织单元（OU），而非通用容器。也就是说，CN=Domain Controllers是错误的，正确的应该是OU=Domain Controllers。

导致账号创建失败的另一个常见的原因是，要使用的账号名已存在。在这种情况下，就要尝试使用另外的计算机账号名。

14.2.2 定制计算机账号属性与组成员关系

如果只提供一个DN，则其他一些参数将自动设置。组成员关系会进行设置，以便该计算机成为Domain Computers的成员。SAM账号名将从计算机DN中使用的常用名属性中推导得出。基本上，DSADD COMPUTER命令会在常用名后面加上一个美元符号（\$）作为后缀。前面的实例中，常用名为CORPSEVER05，因此，其对应的SAM账号名为CORPSEVER05\$。

在创建计算机账号时，如果需要对计算机账号属性进行定制，可以使用如下一些附加的参数。

- -Samid。用于设置SAM账号名，必须以\$结尾，比如，-samid CORPSEVER05\$。
- -Desc。用于设置待添加计算机的描述信息，比如，-desc "CNMember Server"。
- -Loc。用于设置待添加计算机物理位置的文本描述信息。典型情况下，位置信息应该是计算机所在办公室或建筑。比如，如果计算机所在物理位置是建筑物E的110办公室，则可以键入-loc "E/110"。

如果需要为新计算机账号设置组成员关系，可以使用-Memberof参数。这一参数可以接受空格分

隔开的DN（代表新计算机账号所属的组）列表。比如，如果需要将新计算机账号作为Engineering组的成员，该组的DN为CN=Engineering,OU=Eng,DC=cpandl,DC=com，则可以使用类似于如下的命令：

```
dsadd computer "CN=CORPSEVER05,OU=Domain Controllers,DC=cpandl,DC=com"
-memberof "CN=Engineering,OU=Eng,DC=cpandl,DC=com"
```

如果要将新计算机账号作为Engineering组与Tech组的成员，两个组的DN分别为CN=Engineering,OU=Eng,DC=cpandl,DC=com与CN=Tech,CN=Users,DC=cpandl,DC=com，则可以使用类似于如下的命令：

```
dsadd computer "CN=CORPSEVER05,OU=Domain Controllers,DC=cpandl,DC=com"
-memberof "CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Tech,CN=Users,DC=cpandl,DC=com"
```

注解 没有必要指定Domain Computers作为组，新计算机账号会自动成为该组（以及其他指定组）的成员。

14.3 管理计算机账号属性

与使用活动目录用户和计算机相比，在命令行中对计算机账号进行管理有些微不同，主要是因为有更多的选项，尤其是涉及同时操作多个计算机账号时。

14.3.1 查看与寻找计算机账号

第13章曾经讲解过，你可以使用DSQUERY COMPUTER命令搜索计算机。不仅可以根据活动目录账号名、SAM账号名、描述信息等进行搜索，还可以在这些字段中使用通配符帮助匹配。DSQUERY COMPUTER命令的输出信息中包含符合搜索标准的计算机的DN，并可以通过管道传递给其他命令，包括DSGET COMPUTER等，并可用于显示计算机账号相关属性。

DSGET COMPUTER最适合与DSQUERY COMPUTER结合使用。其中，DSQUERY COMPUTER用于获取一台或多台计算机的DN，之后使用DSGET COMPUTER显示相关账号的属性，可以显示的属性是使用搜索参数设置的。

- -Dn。在输出信息中显示匹配的计算机账号的DN。
- -Samid。在输出信息中显示匹配的计算机账号的SAM账号名。
- -Sid。在输出信息中显示匹配的计算机账号的安全标识符。
- -Desc。在输出信息中显示匹配的计算机账号的描述信息。
- -Loc。在输出信息中显示匹配的计算机账号的位置属性。
- -Disabled。显示Yes/No，表示计算机账号是否禁用。

DSGET COMPUTER以表格形式显示输出信息。一般来说，总是要使用-Dn、-Samid、-Sid等参数，以便清晰地理解与识别输出信息中的计算机。比如，如果需要搜索所有已禁用的engineering计算机，可以使用如下命令：

```
dsquery computer -name engcomp* | dsget computer -dn -disabled
```

这一命令将显示DN与禁用状态：

```
dn                                disabled
```

```

CN=engcomp18,OU=Eng,DC=cpand1,DC=com      yes
CN=engcomp19,OU=Eng,DC=cpand1,DC=com      yes
CN=engcomp20,OU=Eng,DC=cpand1,DC=com      no
CN=engcomp21,OU=Eng,DC=cpand1,DC=com      no
CN=engcomp22,OU=Eng,DC=cpand1,DC=com      no
dsget succeeded

```

你也可以显示SAM账号名，如下所示：

```

dsquery computer -name engcomp* | dsget computer -samid -disabled
    samid          disabled
    ENGCOMP18$     yes
    ENGCOMP19$     yes
    ENGCOMP20$     no
    ENGCOMP21$     no
    ENGCOMP22$     no
dsget succeeded

```

或者显示安全标识符：

```

dsquery computer -name engcomp* | dsget computer -sid -disabled

    sid                                                    disabled
    S-1-5-21-4087030303-3274042965-2323426166-1119     yes
    S-1-5-21-4087030303-3274042965-2323426166-1120     yes
    S-1-5-21-4087030303-3274042965-2323426166-1122     no
    S-1-5-21-4087030303-3274042965-2323426166-1123     no
    S-1-5-21-4087030303-3274042965-2323426166-1124     no
dsget succeeded

```

无论哪种方式，都可以获取区分计算机账号条目的标识符。你也可以使用DSGET COMPUTER命令的第2种语法格式，来获取计算机账号的组成员关系。比如，如果想了解ENGCOMP18属于哪一个组，你可以键入如下命令：

```
dsquery computer -name engcomp18 | dsget computer -memberof
```

或

```
dsget computer "CN=engcomp18,OU=Eng,DC=cpand1,DC=com" -memberof
```

上面两条命令的工作方式是一样的。第一个实例中，使用DSQUERY COMPUTER获取计算机账号的DN。第二个实例中，直接指定计算机账号的DN。不管哪一条命令，其输出信息都应该展示组成员关系，比如：

```

"CN=Tech,CN=Users,DC=cpand1,DC=com"
"CN=Engineering,OU=Eng,DC=cpand1,DC=com"
"CN=Domain Computers,CN=Users,DC=cpand1,DC=com"

```

输出信息表明，该计算机账号是Tech、Engineering、Domain Computers等组的成员。

虽然这一技术可以用于显示多台计算机的组成员关系，但无法显示相关联的计算机的DN或SAM账号名。因而，你可以获取一个组成员关系列表，但唯一用于表明这些组成员关系适用于不同计算机的标识是输出信息中的空白行。比如，使用如下查询：

```
dsquery computer -name engcomp* | dsget computer -memberof
```

其输出信息类似于如下：


```
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"
```

```
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
```

```
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"
```

```
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"
```

```
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"
```

```
"CN=Tech,CN=Users,DC=cpandl,DC=com"
```

```
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
```

```
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"
```

这里，显示了5个计算机账号的输出信息（输出信息中包含了5个空白行，用于分隔不同的组），但不能显示不同条目相关的具体计算机账号。

真实场景 不要忽略使用DSQUERY COMPUTER总结当前计算机账号配置信息的重要性。下面给出了一个用于总结计算机账号配置信息的命令实例：

```
dsquery computer "DC=cpandl,DC=com" | dsget computer -dn  
-samid -sid -desc -loc -disabled > domaincomputers.txt
```

该命令用于列出cpandl.com域内的所有计算机账号及其属性，并将其存储到文件中。

14.3.2 设置或修改计算机的位置与描述信息属性

在命令行中，使用DSMOD COMPUTER命令可以快速而方便地设置或修改计算机账号的位置与描述信息。实际上，你可以同时为1台、10台、100台或更多的计算机设置位置或描述信息。假定对Engineering OU内的所有500台计算机，希望其描述信息均为Engineering Computer，位置均为Engineering Dept，则可以使用如下命令：

```
dsquery computer "OU=Engineering,DC=cpandl,DC=com" | dsmod computer  
-loc "Engineering Dept." -desc "Engineering Computer"
```

DSMOD COMPUTER将逐一报告每一变更的成功或失败：

```
dsmod succeeded:CN=Engineeringcomp01,OU=Engineering,DC=cpandl,DC=com  
dsmod succeeded:CN=Engineeringcomp02,OU=Engineering,DC=cpandl,DC=com  
dsmod succeeded:CN=Engineeringcomp03,OU=Engineering,DC=cpandl,DC=com  
...  
dsmod succeeded:CN=Engineeringcomp499,OU=Engineering,DC=cpandl,DC=com  
dsmod succeeded:CN=Engineeringcomp500,OU=Engineering,DC=cpandl,DC=com
```

在GUI中，做这些修改可能会花费数小时的时间，但在命令行中则只需要几分钟，只要键入正确的命令，DSMOD COMPUTER将为你完成一切工作。

14.3.3 禁用与激活计算机账号

在命令行中，你可以使用DSMOD COMPUTER命令与-Disabled参数来激活或禁用计算机账号。键入**-disabled yes**用于禁用计算机账号；**-disabled no**则用于激活计算机账号。

下面的实例中，禁用了TestLab OU中的所有计算机账号：

```
dsquery computer "OU=TestLab,DC=cpandl,DC=com" | dsmod computer
-disabled yes
```

DSMOD COMPUTER命令会逐一报告每个计算机账号变更的成功或失败:

```
dsmod succeeded:CN=TestLabcomp01,OU=TestLab,DC=cpandl,DC=com
dsmod succeeded:CN=TestLabcomp02,OU=TestLab,DC=cpandl,DC=com
dsmod succeeded:CN=TestLabcomp03,OU=TestLab,DC=cpandl,DC=com
```

14.3.4 重置锁定的计算机账号

与用户账号类似的是,计算机账号也有相应的口令。与用户账号不同的是,计算机账号的口令是自动管理与维护的。计算机账号有两个口令:一个标准口令,默认情况下每隔30天改变一次;一个私钥口令,用于与域控制器建立安全通信,默认情况下也是每隔30天改变一次。

这两个口令必须同步。如果私钥口令与计算机账号口令之间的同步消失,则该计算机将不被允许登录该域,Netlogon服务也将产生域认证错误消息,事件ID为3210或5722。如果出现这种情况,则计算机账号口令会被认为陈旧的。此时,你可能需要重置该账号,来保持口令同步。

对工作站与服务器,使用的技术是不同的。要重置工作站上失去同步的口令,可以使用DSMOD COMPUTER命令与-Reset参数,参考如下实例:

```
dsmod computer
"CN=Engineeringcomp01,OU=Engineering,DC=cpandl,DC=com" -reset
```

这一命令对cpandl.com域内的Engineering组织单元中的计算机Engineeringcomp01的口令进行了重置。

你也可以方便地重置Engineering OU中所有计算机账号的口令。为此,可以结合使用DSQUERY COMPUTER与DSMOD COMPUTER命令,前者用于获取域内所有计算机列表,后者用于重置口令,比如:

```
dsquery computer "OU=Engineering,DC=cpandl,DC=com" | dsmod computer
-reset
```

真实场景 要确定计算机账号是否包含陈旧口令,可以使用DSQUERY COMPUTER命令与-Stalepwd参数。如果使用的是默认的30天改变一次口令,则可以通过如下命令寻找陈旧口令,如下所示:

```
dsquery computer -stalepwd 30
```

输出信息中将展示超过30天未修改口令的计算机列表。也就是说,口令是陈旧的,或者计算机处于不活跃状态。

对成员服务器与域控制器,应该使用NETDOM RESETPWD命令,而非DSMOD COMPUTER -RESET命令。通过如下步骤,可以实现对成员服务器或域控制器的计算机账号口令进行重置。

(1) 本地登录服务器。如果需要重置域控制器的口令,必须停止Kerberos Key Distribution Center服务,并将其启动方式设置为手动。

(2) 在增强的命令提示符中,键入命令netdom resetpwd /s: ComputerName /ud:domain\user /pd:*。其中,ComputerName为该计算机账号登录域内的域控制器名,domain\user为管理员账号名,具备修改计算机账号口令的权限,*的作用是使得NETDOM在继续执行之前弹出提示信息(要求输入账号口令)。

(3) 输入口令后, NETDOM会在本地与适当的域控制器上修改计算机账号口令。该域控制器会向其他域控制器分发口令变更情况。

(4) NETDOM完成任务后, 重启计算机并确认口令已成功重置。如果重置的是域控制器的口令, 则需要重启Kerberos Key Distribution Center服务, 并将其启动方式设置为自动。

真实场景

作为故障排除过程的一部分, 你应该总是注意对活动目录中的计算机账号状态进行检查, 包括计算机的存储位置及其组成员关系。与用户账号类似, 计算机账号被放置到活动目录的特定容器中, 也可以被添加为特定组的成员。计算机账号所在容器决定了活动目录策略设置如何作用于该计算机。将计算机移动到不同的容器或OU后, 组策略设置作用于该计算机的方式也将有所改变。

你也应该检查组员关系。计算机的组成员关系决定了该计算机的多种许可权限, 比如安全性与资源访问等。改变组成员关系也会影响该计算机的许可权限。如果使用Kerberos认证方式, 计算机的系统时间对认证也会有影响。如果计算机的系统时间偏离了组策略中允许范围内的值, 则该计算机的认证将会失败。

14.3.5 将计算机账号添加到某域中

认证后的任意用户都可以将计算机添加到某个域中, 这需要使用NETDOM JOIN命令。如果相关的计算机账号尚未创建, 运行这一命令将创建该账号。计算机添加到某个域后, 就与该域建立了信任关系。计算机的安全标识符也将被修改, 以便与活动目录中相关计算机账号匹配, 该计算机也将成为活动目录中某个适当组的成员。典型情况下, 计算机将成为Domain Computers组的成员。如果该计算机后来被用作域控制器, 则该计算机将变更为Domain Controllers组的成员。

注解 NETDOM在Windows Server 2008上是可用的, 但在Windows Vista上是不可用的(除非自己安装)。在将计算机添加到域之前, 应该检查计算机的网络配置。如果网络配置不正确, 在添加到域之前, 应该修正它。此外, 如果该计算机账号是以前创建的, 则只有具备专门许可权限的用户或管理员才可以将该计算机添加到某个域中。用户必须具备本地计算机上的本地管理员权限。

你可以通过两种方式将计算机添加到某个域中。

- 登录到需要添加到某个域的计算机, 之后运行NETDOM JOIN命令。
- 从其他计算机上运行NETDOM JOIN命令, 连接到需要添加到某个域的计算机。

登录到需要添加到某个域的计算机后, 可以使用NETDOM JOIN命令将计算机添加到某个域, 同时在该域内创建计算机账号, 这是使用如下的命令语法实现的:

```
netdom join ComputerName /Domain:DomainName /UserD: DomainUser
/PasswordD: UserPassword
```

其中, ComputerName为计算机名, DomainName为要加入的活动目录域名, DomainUser为将计算机加入到该域的授权的域用户账号(如果必要, 还要创建相关的计算机账号), UserPassword为该用户账号的口令。域用户账号应该以DOMAIN\UserName的形式指定, 比如CPANDL\WilliamS。参考如下实例:

```
netdom join corpsvr32 /domain:cpandl.com /userd:CPANDL\williams
/passwordd:2892389383234
```

上面的命令中，将CorpSvr32添加到cpandl.com域中，并在默认的Computers容器内创建了相关的计算机账号。该计算机对象的全路径为CN=CorpSvr32,CN=Computers,DC=cpandl,DC=com。

此外，你也可以使用/Ou参数指定OU的区分名（计算机应该放置到该OU中）。如果需要关机，并在加入到域后重启计算机，则可以使用/Reboot参数。参考如下实例：

```
netdom join corpsvr32 /domain:cpandl.com
/ou:OU=Engineering,DC=cpandl,DC=com
/userd:CPANDL\williams /passwordd:2892389383234 /reboot
```

上面的命令中，将CorpSvr32添加到cpandl.com域中，并在Engineering OU内创建了相关的计算机账号，之后重新引导计算机。该计算机对象的全路径为CN=CorpSvr32,OU=Engineering, DC=cpandl, DC=com。

从其他计算机运行NETDOM JOIN并连接到待加入域的计算机时，可以使用如下的命令语法格式：

```
netdom join ComputerName /Domain:DomainName /UserD: DomainUser
/PasswordD: UserPassword /UserO: ComputerUser
/PasswordO: ComputerPassword
```

其中，*ComputerUser*为要加入域内的计算机上的本地用户账号名，*ComputerPassword*为该用户账号的密码。和前面的其他命令类似，这一命令也将创建相关的计算机账号（如果必要）。此外，你还可以使用/Ou参数（可选地），用来指定该计算机将放置进去的OU的区分名。

参考如下实例：

```
netdom join desktop267 /domain:cpandl.com /ou:OU=Services,DC=cpandl,DC=com
/userd:CPANDL\williams /passwordd:2892389383234 /usero:sammiep
/passwordo:383478478722
```

上面的命令中，将Desktop267添加到cpandl.com域中，并在Server OU内创建了相关的计算机账号。该计算机对象的全路径为CN=Desktop267, OU=Services,DC=cpandl,DC=com。

14.3.6 对计算机与计算机账号进行重命名

通过NETDOM RENAMECOMPUTER命令，可以很方便地对工作站与成员服务器进行重命名。如果工作站或成员服务器加入到了域中，则相关的计算机账号也会重命名。然而，你不能使用NETDOM RENAMECOMPUTER命令对如下一些计算机进行重命名，包括域控制器，运行证书服务的服务器，运行其他需要特定的，固定的服务器名的服务器。

本地登录后，可以使用如下的命令语法对工作站或成员服务器进行重命名：

```
netdom renamecomputer ComputerName /NewName:NewName /UserD:DomainUser
/PasswordD:UserPassword
```

其中，*ComputerName*为计算机的当前名，*NewName*为计算机的新名，*DomainUser*为将计算机进行重命名的授权的域用户账号，*UserPassword*为该用户账号的口令。域用户账号应该以DOMAIN\UserName的形式指定，比如CPANDL\WilliamS。参考如下实例：

```
netdom renamecomputer desktop16 /newname:wkstn75 /userd:CPANDL\williams
/passwordd:2892389383234
```

上面的命令中，将Desktop16重命名为Wkstn75，这是使用WilliamS账号完成的。如果需要关机，

并在重命名后重启计算机，则可以使用/Reboot参数。参考如下实例：

从其他计算机运行NETDOM RENAMECOMPUTER命令并连接到欲重命名的计算机时，可以使用如下的命令语法格式：

```
netdom renamecomputer ComputerName /NewName:NewName /UserD:DomainUser
/PasswordD:UserPassword /UserO:ComputerUser /PasswordO:ComputerPassword
```

其中，*ComputerUser*为授权连接到待重命名计算机的用户账号名，*ComputerPassword*为该计算机账号的口令。参考如下实例：

```
netdom renamecomputer desktop143 /newname:wkstn76 /userd:CPANDL\williams
/passwordd:2892389383234 /usero:sammiep /passwordo:383478478722
```

上面的命令中，将Desktop143重命名为Wkstn76，这是使用WilliamS账号完成的。如果需要关机，并使用SammieP账号连接到待重命名的计算机。

14.3.7 移动计算机账号

计算机账号通常放置在Computers、Domain Controllers或自定义的OU容器中。通过DSMOVE命令，可以将计算机账号移动到当前域内的其他位置或OU中。指定计算机账号的当前DN，之后使用-Newparent参数指定该计算机账号的新位置或父DN。如果需要将计算机账号CORPSVR03从Tech OU移动到Engineering OU，你需要指定该计算机账号的DN，比如“CN=CORPSVR03,OU=Tech,DC=cpandl,DC=com”。并提供新位置的父DN，比如“OU=Engineering,DC=cpandl,DC=com”，相关命令类似于如下的格式：

```
dsmove "CN=CORPSVR03,OU=Tech,DC=cpandl,DC=com"
-newparent "OU=Engineering,DC=cpandl,DC=com"
```

这里假定已经使用DSQUERY COMPUTER命令获取了计算机账号DN，可以将DSQUERY COMPUTER的输出通过管道传递给DSMOVE，如下面实例所示：

```
dsquery computer -name "CORPSVR03" | dsmove
-newparent "OU=Engineering,DC=cpandl,DC=com"
```

其中，计算机账号DN“CN=CORPSVR03,OU=Tech,DC=cpandl,DC=com”是通过DSQUERY COMPUTER命令获取的，并用作DSMOVE的输入信息。不管计算机账号是工作站、成员服务器还是域控制器，上面的命令都是有效的。

14.3.8 删除计算机账号

如果不再需要某个计算机账号，可以使用DSRM命令将其从活动目录中永久删除。大多数情况下，你可能需要删除某个特定的计算机账号，比如Corpserver03。如果是这种情况，你可以将该计算机账号的DN传递给DSRM命令，比如：

```
dsrcm "CN=corpserver03,OU=Eng,DC=cpandl,DC=com"
```

默认情况下，DSRM会弹出提示信息询问是否确认删除。如果不希望看到提示信息，可以使用-Noprompt参数，比如：

```
dsrcm "CN=corpserver03,OU=Eng,DC=cpandl,DC=com" -noprompt
```


14.4 操作域控制器

运行Windows Server 2008的计算机可以充当成员服务器或域控制器。本章前面讲述的技术和方法适用于任意类型的计算机账号，但本节讲述的内容只适用于域控制器。

14.4.1 安装与降级域控制器

Windows Server 2008支持两种类型的域控制器，可读写的域控制器（RWDC）与只读的域控制器（RODC）。RWDC存储活动目录中可写的复制数据，RODC存储活动目录中只读的复制数据。

在活动目录域中，域控制器执行很多重要的任务。通过使用**servermanagercmd -install adds-domain-controller**安装Active Directory Domain Services角色，之后运行DCPROMO命令，可以安装目录服务，并将成员服务器升级为域控制器。如果再一次运行DCPROMO命令，就会对该域控制器进行降级，使其再一次只是充当成员服务器的角色。

真实场景 DCPROMO命令会启动一个图形界面工具，但也可以接受几个命令行参数，包括/Answer:FileName与/Adv。通过/Answer参数，可以提供一个应答文件，用于脚本化目录服务安装。如果需要对整个服务器进行自动安装，则应该在Unattend.txt文件中添加一个GUIRunOnce条目，以便于在无人值守安装后启动DCPROMO。通过/Adv参数，DCPROMO会以高级模式运行，该模式下提供了很多选项，可以在安装时创建域控制器，或者从安装介质备份中创建。对这两种方式，要复制域信息，首先都需要为运行Windows Server 2008的域控制器（运行在同一域内，是待升级的成员服务器）创建安装介质或介质备份，之后使用DCPROMO启动活动目录的高级安装。

14.4.2 在活动目录中发现域控制器

如果需要操作域控制器，而不是所有计算机账号，可以使用DSQUERY SERVER命令与DSGET SERVER命令。默认情况下，使用DSQUERY SERVER命令时，搜索的是所在的登录域。实际上，只键入**dsquery server**命令并按Enter键后，会返回一个列表，其中包含了所在登录域内的所有域控制器。如果必要，可以使用-Domain参数指定待搜索的域，参考如下实例：

```
dsquery server -domain tech.cpandl.com
```

这一命令执行后，会获取tech.cpandl.com域内的所有域控制器列表。如果需要获取的是整个森林内的所有域控制器列表，可以使用命令**dsquery server -forest**。

在所有这些实例中，输出信息都是域控制器DN列表。但与前面讲述的DN不同的是，这里的DN还包含了站点配置信息，比如：

```
"CN=CORPSVR02, CN=Servers, CN=Default-First-Site-Name, CN=Sites, CN=Configuration, DC=cpandl, DC=com"
```

这些附加的信息是由DSQUERY SERVER命令提供的，用于指定与服务器相关联的站点。要记住的是，域可以跨越不止一个物理位置，而告知活动目录这一点的方法是使用站点与子网。上面的实例中，相关联的站点是站点配置容器中的Default-First-Site-Name。

有时候，你可能需要发现特定站点中的域控制器，为此可以使用-site参数指定站点。下面的实例中，

试图发现Seattle-First-Site站点中的所有域控制器：

```
dsquery server -site Seattle-First-Site
```

注解 DSQUERY SERVER命令有一些附加的参数，可以用于搜索全局编目、操作主机以及只读的域控制器。这些参数将分别在14.5.1、14.6.1、14.7等节进行讨论。

与其他计算机相关的命令类似，DSQUERY SERVER命令与DSGET SERVER命令适合于一起使用。DSQUERY SERVER命令用于获取一个或多个域控制器的DN，DSGET SERVER命令用于显示相关账号的属性，可以显示的属性是使用如下参数指定的。

- -Dn。显示输出信息中匹配域控制器的DN。
- -Desc。显示输出信息中匹配域控制器的描述信息。
- -Dnsname。显示域控制器的完全限定域名。
- -Is_gc。显示Yes/No值，用于表示域控制器是否同时还是一个全局编目服务器。

比如，如果需要获取森林中所有域控制器的详细摘要，可以键入如下命令：

```
dsquery server -forest | dsget server -desc -dnsname -is_gc
```

要保存这些信息，可以将其重定向到文件中，比如：

```
dsquery server -forest | dsget server -desc -dnsname -is_gc >  
forest-dcs.txt
```

14.5 指定全局编目服务器

指定为全局编目服务器的域控制器中存储了活动目录中所有对象的完全副本（用于其主机域）以及域森林中所有其他域的部分副本。全局编目用于登录与信息搜索。实际上，如果全局编目服务不可用，则正常用户也将无法登录到相应域。要防止这种情况，唯一的方法是在本地域控制器上缓存通常的组成员关系信息。默认情况下，域内安装的第一个域控制器将被指定为全局编目服务器。你也可以向域中添加全局编目服务，来提高登录与查询请求的响应速度。建议为域内每个站点都配置一个全局编目服务器。

任意提供全局编目的域控制器都应该与网络以及充当架构主机的域控制器建立稳定的连接，架构主机是可以为域控制器指定的5种操作主机之一。其作用是负责对象引用信息的更新，这是通过将数据与全局编目中的数据进行比较实现的。如果架构主机发现陈旧数据，就会从全局编目中请求陈旧数据，之后向域内的其他域控制器复制数据所做的更改。

提示 如果域中只有一个域控制器，可以为同一个域控制器同时指定架构主机角色与全局编目服务。如果域中有两个或多个域控制器，架构主机角色与全局编目服务就不应该指定在同一台域控制器上，因为这会影响基础设施主机判断目录数据是否过期的准确性。

14.5.1 发现全局编目服务器

如果需要了解全局编目服务的部署位置，对当前（登录）域，只需键入dsquery server -is_gc命令，输出信息中将包含全局编目的DN，比如：

命名主机负责域森林中域的添加与移除。由于在域森林中，这两个角色都是唯一的，因此只能指定一台架构主机与一台域命名主机。

另外几种必须指定的域角色是相对ID主机、PDC模拟器主机与基础结构主机。从名字可以看出，相对ID主机为域控制器分配相对ID。创建用户、组或计算机对象时，域控制器会为相关对象指定一个唯一性的安全ID。安全ID包含了域安全ID前缀以及一个唯一的相对ID，后者就是由相对ID主机进行分配的。PDC模拟器主机负责处理用户、计算机、trusts的口令更改。基础结构主机负责对对象索引进行更新，这是通过对目录数据与全局编目中的数据进行比较实现的。如果基础结构主机发现陈旧数据，就会从全局编目中请求陈旧数据，之后向域内的其他域控制器复制数据所做的更改。在每个域中，这些作用于域森林范围的角色必须是独一无二的。也就是说，在每个域内，只可以指定一台相对ID主机、一台PDC模拟器主机、一台基础结构主机。

14.6.1 发现操作主机

安装新网络时，第一个域内的第一个域控制器会被指定为所有的操作主机角色。如果后来在新树中创建子域或根域，其中的第一个域控制器也将会被自动地指定为这些操作主机角色。在一个新的域森林中，域控制器会被指定为所有操作主机角色。如果新建域在原有的域森林中，则指定的角色包括相对ID主机、PDC模拟器主机与基础结构主机。架构主机与域命名主机则仍保留在森林的第一个域内。如果必要，管理员可以变更操作主机角色。

通过如下命令，可以确定所在登录域内的当前操作主机：

```
netdom query fsmo
```

输出信息中列出了每个角色的属主，且以完全限定域名描述：

Schema master	CorpServer18.cpandl.com
Domain naming master	CorpServer35.cpandl.com
PDC	CorpServer23.eng.cpandl.com
RID pool manager	CorpServer23.eng.cpandl.com
Infrastructure master	CorpServer49.eng.cpandl.com

从输出信息中可以确定，森林根域为cpandl.com，当前登录域为eng.cpandl.com。如果需要确定某个特定域的操作主机，可以使用如下命令：

```
netdom query fsmo /d:DomainName
```

其中，*DomainName*为域名，比如eng.cpandl.com。

通过在DSQUERY SERVER命令中使用-Hasfsmo参数，可以确定森林中的哪个域控制器或域指定了操作主机角色。这一参数有如下一些取值。

- **scheme**。返回森林的架构主机DN。
- **name**。返回森林的域命名主机的DN。
- **infr**。返回森林的架构主机DN。如果没有使用-**Domain**参数指定域，则默认使用当前域。
- **pdcc**。返回森林的PDC模拟器主机DN。如果没有使用-**Domain**参数指定域，则默认使用当前域。
- **rid**。返回森林的相对ID主机DN。如果没有使用-**Domain**参数指定域，则默认使用当前域。

架构主机与域命名主机是森林范围的角色。键入dsquery server -hasfsmo scheme或dsquery server -hasfsmo name命令时，总是会获取活动目录森林内相关操作主机的DN。

相对ID主机、PDC模拟器主机与基础结构主机是域范围的角色。键入dsquery server -hasfsmo infr、

dsquery server -hasfsmo pdc或**dsquery server -hasfsmo rid**命令时，总是会获取登录域内相关操作主机的DN。如果需要获取其他域内操作主机的DN，则可以使用-Domain参数，参考如下实例：

```
dsquery server -hasfsmo rid -domain tech.cpandl.com
```

上面的命令将获取tech.cpandl.com域内相对ID主机的DN。如果森林中包含了多个域，你可能需要获取所有域控制器列表，其中每个域控制器都具备一个以该域为基础的特定角色。为此，可以使用-Forest参数，比如：

```
dsquery server -hasfsmo rid -forest
```

14.6.2 使用命令行配置操作主机角色

尽管可以使用目录服务命令检查操作主机所在位置，但不能用于对操作主机角色进行配置。如果需要对操作主机角色进行配置，必须使用NTDSUtil，NTDSUtil是一个文本模式的命令解释器，通过调用NTDSUtil，可以使用单独的命令提示符与内部命令对目录服务进行管理。要调用NTDSUtil解释器，可以在命令窗口中键入ntdsutil，之后按Enter键。

通过NTDSUtil，可以将操作主机角色从一个域控制器迁移到另外的域控制器，也可以在角色不能正常迁移时对其进行查封。比如，充当基础结构主机的域控制器发生了磁盘失效，使得整个服务器不可用。如果无法使得该服务器联机，就可能需要查封这一基础结构主机角色并将该角色赋予其他域控制器。对于以后还需要联机的域控制器，切记永远不要查封其角色。一旦查封了角色，则服务器将再也无法恢复服务。要使得原始服务器主机联机，唯一的方法是格式化引导磁盘，并重装Windows Server 2008。因此，对角色进行查封要慎之又慎，通常是万不得已的选择。

在对角色进行查封并强制迁移之前，应该确定接管该角色的域控制器相对于角色原属主的更新程度。活动目录使用更新序列号（USN）对数据变更情况进行追踪。由于数据复制存在延迟，因此不是所有域控制器上都保持了最新数据。如果将域控制器的USN与域内其他服务器进行比较，就可以确定域控制器与角色原属主相比，数据是否是最新的。如果域控制器是最新的，就可以安全地迁移角色。如果域控制器数据不是最新的，就需要等待数据复制的进行，之后将角色迁移到该域控制器。

Windows Server 2008包含了REPADMIN命令，用于操作活动目录复制。如果需要显示指定域控制器中每个复制方特定命名上下文中的最大序列号，可以使用如下命令：

```
repadmin /showutdvec DomainControllerName NamingContext
```

其中，DomainControllerName是域控制器的完全限定域名，NamingContext为服务器所在域的区分名，比如：

```
repadmin /showutdvec corpserver45 dc=cpandl,dc=com
```

上面命令的输出信息将会展示域分区中复制方的最大USN：

```
Main-Site\corpserver18 @ USN 967382 @ Time 2008-05-15 10:12:22
Main-Site\corpserver92 @ USN 970043 @ Time 2008-05-15 10:12:25
```

这一实例中，如果CorpServer18是以前的角色属主，并且正在检查的域控制器的USN大于或等于CorpServer18的USN，则说明域控制器是最近更新的；如果CorpServer18是以前的角色属主，并且正在检查的域控制器的USN小于CorpServer18的USN，则说明域控制器不是最新更新的。对这种情况，在对角色进行查封之前，应该等待数据复制的进行。你可以使用REPADMIN/SYNCALL强制最近更新的

```
"CN=CORPSVR02,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com"
```

DSQUERY SERVER也可以用于定位特定域中的全局编目。要做到这一点,需要使用-Domain参数,比如键入命令**dsquery server -domain tech.cpandl.com -isgc**。

这一命令对tech. cpandl.com域内的全局编目服务器进行搜索。如果需要对整个森林内的全局编目服务器进行搜索,可以使用命令**dsquery server -forest -isgc**。

你也可以根据站点搜索全局编目服务器,但要做到这一点,必须知道完整的站点名,而不能使用通配符。比如,如果需要发现Default-First-Site-Name的所有全局编目服务器,可以键入命令**dsquery server -site Default-First- Site-Name**。

注解 可以根据站点搜索全局编目服务器是一项重要的功能,因为典型情况下每个站点都有一台全局编目服务器。如果搜索一个站点但未能发现全局编目服务,则可以考虑添加一个。

14.5.2 添加或移除全局编目服务器

通过DSMOD SERVER命令,其后跟随待操作服务器的DN,之后使用**-isgc yes**参数,就可以将域控制器指定为全局编目服务器,比如:

```
dsmod server "CN=corpdc05,OU=Eng,DC=cpandl,DC=com" -isgc yes
```

完成这一任务的另一种方式是使用DSQUERY SERVER命令获取待操作服务器的列表。假定tech. cpandl.com域包含3台域控制器,现在需要将其全部指定为全局编目,则可以使用如下命令:

```
dsquery server -domain tech.cpandl.com | dsmod server -isgc yes
```

上面的命令中,使用DSQUERY SERVER命令获取tech. cpandl.com域内所有域控制器的DN,之后将这些信息传递给DSMOD SERVER命令,之后该命令将其全部指定为全局编目。

如果后来需要某台服务器停止全局编目服务,可以使用**-isgc no**参数。下面的实例中,不再需要tech. cpandl.com域内的corpdc04服务器提供全局编目服务:

```
dsmod server "CN=corpdc04,OU=Tech,DC=cpandl,DC=com" -isgc no
```

14.5.3 检查缓存与优先的全局编目设置

域与域森林的不同功能层级依赖于网络配置。如果域或域森林内所有域控制器运行的至少是Windows 2000 Server以上版本,功能层级设置为Windows 2000本原模式,则组织可以充分利用活动目录很多附加的功能,但不再能使用Windows NT的主域控制器(PDC)与备份域控制器(BDC)。Windows 2000本原模式下激活的一个功能是通常组成员关系信息缓存。

如果用户尝试登录但域内不存在全局编目,就可能导致无法登录,但通过通常组成员关系信息缓存,就可以解决这一问题。这种缓存功能的禁用或激活是以每个站点为基础的,要确定某个站点的这种缓存功能是否激活,可以使用DSGET SITE命令,将待操作站点DN传递给该命令,其后跟随-Cachegroups参数,如下面实例所示:

```
dsget site "CN=Default-First-Site-
Name, CN=Sites,CN=Configuration,DC=cpandl,DC=com" -cachegroups
```


如果通常组成员关系信息缓存是激活的，则输出信息类似于如下的格式：

```
cacheGroups
是
dsget 成功
```

如果通常组成员关系信息缓存是禁用的，则输出信息类似于如下的格式：

```
cacheGroups
否
dsget 成功
```

完成这一搜索的另一种方式是使用DSQUERY SITE命令。在命令行中键入**dsquery site**命令，不带任何参数，则该命令会返回一个列表，包含了域森林内所有站点。要对返回的结果集进行限制，可以使用**-Name**参数指定站点的通常名，或使用通配符指定部分名，比如：

```
dsquery site -name *First*
```

这一命令搜索通常名中包含First的所有站点。

将上面的方法结合在一起，你可以使用如下命令确定域森林中所有站点的缓存设置：

```
dsquery site | dsget site -cacheGroups
```

这一命令的输出信息中包含了一些“是”或“否”的应答信息，类似于如下格式：

```
cacheGroups
是
是
否
否
是
dsget 成功
```

如果希望输出信息具有更丰富的含义，可以使用**-Dn**参数，来显示相关站点的DN，比如：

dn	cacheGroups
CN=Seattle-Site-Name,CN=Sites,CN=Configuration,DC=cpand1,DC=com	是
CN=LA-Site-Name,CN=Sites,CN=Configuration,DC=cpand1,DC=com	是
CN=NY-Site-Name,CN=Sites,CN=Configuration,DC=cpand1,DC=com	是
CN=Chicago-Site-Name,CN=Sites,CN=Configuration,DC=cpand1,DC=com	是
CN=Detroit-Site-Name,CN=Sites,CN=Configuration,DC=cpand1,DC=com	是

```
dsget 成功
```

如果通常组成员关系信息缓存是激活的，则带有多个全局编目服务（每个站点一个）的域可以有优先的全局编目。这一优先的全局编目专门用于为站点的域控制器刷新通常组成员关系信息缓存。你可以使用**-PrefgcSite**参数来确定优先的全局编目。比如，键入**dsquery site | dsget site -cache groups -prefgcSite**命令，就可以获取域森林中所有全局编目的完整缓存配置信息。如果已经配置了优先的全局编目，会在返回信息中看到“是”或“否”等值。如果没有配置，则在返回信息中看到“未配置”值。

14.6 指定操作主机

活动目录中，定义了5种操作主机角色，每一种角色都在确保网络正常运行中发挥关键作用。其中有些角色可以只在域森林中指定，其他角色则必须在每个域中指定。

作用于域森林范围的角色包括架构主机与域命名主机，架构主机负责目录架构的更新与修改，域

域控制器（与以前的角色属主相比）对所有复制方进行数据复制操作。

在命令行中，通过如下步骤，可以实现角色的迁移。

(1) 登录将要指定新操作主机角色的服务器，启动命令提示符。

(2) 在命令提示符中，键入`ntdsutil`，调用文本模式的命令解释器NTDSUtil。

(3) 在`ntdsutil`提示符中，键入`roles`。这将使得该工具进入操作主机维护模式，提示符也将变为：

`fsmo maintenance:`

(4) 在`fsmo maintenance`提示符中，键入`connections`，以便进入`server connections`提示符，之后键入`connect to server`，其后跟随该角色的当前架构主机的完全限定域名，比如：

`connect to server corpd01.eng.cpandl.com`

(5) 成功建立连接后，键入`quit`，退出`server connections`提示符，之后在`fsmo maintenance`提示符中键入`transfer`，之后再键入待迁移角色的标识符，标识符包括下面5个。

☐ `pdca`-用于PDC模拟主机角色。

☐ `rid`-用于相对ID主机角色。

☐ `infrastructure master`-用于基础结构主机角色。

☐ `scheme master`-用于架构主机角色。

☐ `naming master`-用于域命名主机角色。

(6) 角色迁移完成。在`fsmo maintenance`提示符中键入`quit`，之后在`ntdsutil`提示符中，键入`quit`。

如果不能顺利地实现角色迁移，比如充当该角色的服务器当前处于脱机，或不可用，则可以通过如下步骤查封该角色。

(1) 确信充当待查封角色的当前域控制器已经永久性脱机。如果该服务器还可以联机，不要执行下面的步骤，除非已经计划完全重装该服务器。

(2) 登录将要指定新操作主机角色的服务器，启动命令提示符。

(3) 在命令提示符中，键入`ntdsutil`，调用文本模式的命令解释器NTDSUtil。

(4) 在`ntdsutil`提示符中，键入`roles`。这将使得该工具进入操作主机维护模式，提示符也将变为：

`fsmo maintenance :`

(5) 在`fsmo maintenance`提示符中，键入`connections`，之后在`server connections`提示符中键入`connect to server`，其后跟随该角色的当前架构主机的完全限定域名，比如：

`connect to server corpd01.eng.cpandl.com`

(6) 成功建立连接后，键入`quit`，退出`server connections`提示符，之后在`fsmo maintenance`提示符中键入`seize`，之后再键入待查封角色的标识符，标识符包括下面5个。

☐ `pdca`-用于PDC模拟主机角色。

☐ `rid`-用于相对ID主机角色。

☐ `infrastructure master`-用于基础结构主机角色。

☐ `scheme master`-用于架构主机角色。

☐ `naming master`-用于域命名主机角色。

(7) 角色查封完成。在`fsmo maintenance`提示符中键入`quit`，之后在`ntdsutil`提示符中，键入`quit`。

14.7 发现只读的域控制器

想要确定只读域控制器所在位置？对当前（登录）域，键入`dsquery server -isreadonly`，其输出信息中包含了只读域控制器的DN列表，比如：

```
"CN=CORPSVR48,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com"
```

DSQUERY SERVER命令也可以用于在特定域内定位只读的域控制器。要做到这一点，需要使用`-Domain`参数，比如：

```
dsquery server -domain seattle.cpandl.com -isreadonly
```

上面的命令对seattle.cpandl.com域内的只读域控制器进行搜索。如果需要对整个森林进行搜索，则需要键入`dsquery server -forest -isreadonly`。

你也可以以站点为基础搜索只读域控制器，但要做到这一点，必须知道完整的站点名，而不能使用通配符。比如，如果需要发现Default-First-Site-Name的所有只读域控制器，可以键入命令`dsquery server -site Default-First-Site-Name -isreadonly`。



创建并管理用户与组账号是管理员的核心任务之一。本章首先讲述了如何在命令行中创建并管理用户账号，之后讲述了如何在命令行中创建并管理组账号。操作活动目录用户与组是本章的核心内容。

15.1 从命令行中管理用户账号概览

在Windows Server 2008中，定义了下面两种类型的用户账号。

- **域用户账号**。在活动目录中定义，可以访问整个域内的资源。可以使用目录服务命令创建并管理域用户账号。
- **本地用户账号**。在本地计算机上定义，在访问网络资源之前必须通过认证。可以使用网络服务命令创建并管理本地用户账号。

注解 本地机器账号主要用于工作组配置，而非Windows域。网络上的每台计算机都包含了一个或多个本地机器账号。唯一例外的是域控制器，域控制器上没有配置本地机器账号。如果需要操作本地机器账号，可以使用网络服务命令。

可用于管理域用户账号的目录服务命令包括下面列举的。

- **DSADD USER**。在活动目录中创建用户账号，其语法格式如下：

```
dsadd user UserDN [-samid SAMName] [-upn UPN] [-fn FirstName]
[-mi Initial] [-ln LastName] [-display DisplayName]
[-empid EmployeeID]
[-pwd {Password | *}] [-desc Description] [-memberof Group ...]
[-office Office] [-tel PhoneNumber] [-email EmailAddress]
[-hometel HomePhoneNumber] [-pager PagerNumber]
[-mobile CellPhoneNumber] [-fax FaxNumber] [-iptel IPPhoneNumber]
[-webpg WebPage] [-title Title] [-dept Department]
[-company Company] [-mgr Manager] [-hmdir HomeDirectory]
[-hmdrv DriveLetter:] [-profile ProfilePath] [-loscr ScriptPath]
[-mustchpwd {yes | no}] [-canchpwd {yes | no}]
[-reversiblepwd {yes | no}] [-pwdneverexpires {yes | no}]
[-acctexpires NumberOfDays] [-disabled {yes | no}]
[{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-q] [{-uc | -uco | -uci}]
[-fnp FirstNamePhonetic] [-lnp LastNamePhonetic]
[-displayp DisplayNamePhonetic]
```

- **DSGET USER**。显示用户账号的属性，有两种语法。

显示多个用户账号的语法格式如下：

```
dsget user UserDN ... [-dn] [-samid] [-sid] [-upn] [-fn] [-mi]
[-ln] [-display] [-fnp] [-lnp] [-displayp] [-effectivepso]
[-empid] [-desc] [-office] [-tel] [-email] [-hometel] [-pager]
[-mobile] [-fax] [-iptel] [-webpg] [-title] [-dept] [-company]
[-mgr] [-hmdir] [-hmdrv] [-profile] [-loscr]
[-mustchpwd] [-canchpwd] [-pwdneverexpires] [-disabled]
[-acctexpires] [-reversiblepwd] [{-uc | -uco | -uci}]
[-part PartitionDN [-qlimit] [-qused]] [{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-c] [-q] [-l]
```

查看用户的组成员关系的语法格式如下：

```
dsget user UserDN [-memberof [-expand]] [{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-c] [-q] [-l]
[{-uc | -uco | -uci}]
```

□ **DSMOD USER**。对活动目录中一个或多个用户账号的属性进行修改，其语法格式如下：

```
dsmod user UserDN ... [-upn UPN] [-fn FirstName] [-mi Initial]
[-ln LastName] [-display DisplayName] [-empid EmployeeID]
[-pwd {Password | *}] [-desc Description] [-office Office]
[-tel PhoneNumber] [-email EmailAddress] [-hometel HomePhoneNumber]
[-pager PagerNumber] [-mobile CellPhoneNumber] [-fax FaxNumber]
[-iptel IPPhoneNumber] [-webpg WebPage] [-title Title]
[-dept Department] [-company Company] [-mgr Manager]
[-hmdir HomeDirectory] [-hmdrv DriveLetter:] [-profile ProfilePath]
[-loscr ScriptPath] [-mustchpwd {yes | no}] [-canchpwd {yes | no}]
[-reversiblepwd {yes | no}] [-pwdneverexpires {yes | no}]
[-acctexpires NumberOfDays] [-disabled {yes | no}]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-c] [-q]
[{-uc | -uco | -uci}] [-fnp FirstNamePhonetic]
[-lnp LastNamePhonetic] [-displayp DisplayNamePhonetic]
```

提示 这些命令可以接受来自DSQUERY USER的输入信息，用来为待操作的用户设置区分名（DN）。你也可以为每个用户键入DN，但要确保每个DN之间使用空格分隔。

一瞥之下，这些命令似乎超级复杂，但实际上并非如此复杂——尽管功能非常丰富。通过这些命令，可以添加、查看或修改用户账号，并包含一个可操作的账号属性扩展集。不管是添加、查看还是修改用户账号，用于操作账号某属性的参数是相同的。比如，创建账号时，可以使用-Tel参数设置该用户的办公室电话号码；如果需要确定某个用户的电话号码，可以在DSGET USER命令中使用-Tel参数；如果需要修改某用户的电话号码，可以在DSMOD USER命令中使用-Tel参数。

要对本地机器账号进行管理，可以使用NET USER命令。该命令是一个网络服务命令，具有几种语法格式，具体语法格式依赖于要完成的任务，如下所示。

显示或修改本地用户账号：

```
net user [UserName [Password | *]] [/active:{no | yes}] [/comment:
"DescriptionText"] [/countrycode:NNN] [/expires:{{MM/DD/YYYY |
DD/MM/YYYY | mmm,dd,YYYY} | never}] [/fullname:"Name"]
[/homedir:Path] [/passwordchg:{yes | no}] [/passwordreq:{yes |
no}] [/profilepath:[Path]] [/scriptpath:Path] [/times:{Day[-Day]
[,Day[-Day]] ,Time[-Time] [,Time[-Time]] [;... ] | all}]
```



```
[/usercomment:"Text"] [/workstations:{ComputerName[,...] | *}]
```

创建本地用户账号:

```
net user [UserName {Password | *} /add [/active:{no | yes}]
[/comment:"DescriptionText"] [/countrycode:NNN] [/expires:
{{MM/DD/YYYY | DD/MM/YYYY | mmm,dd,YYYY} | never}] [/fullname:"Name"]
[/homedir:Path] [/passwordchg:{yes | no}] [/passwordreq:{yes | no}]
[/profilepath:[Path]] [/scriptpath:Path] [/times:{Day[-Day]
[,Day[-Day]] ,Time[-Time] [,Time[-Time]] [;...] | all}]
[/usercomment:"Text"] [/workstations:{ComputerName[,...] | *}]
```

删除本地用户账号:

```
net user UserName /delete
```

可以看出, NET USER命令只可以操作相当有限的用户属性集, 这些账号属性适合于操作本地用户账号。

注解 你也可以使用NET USER命令操作登录域内的域控制器, 但对登录域的上层域, 则不具备访问权限, 而目录服务命令则可以在活动目录森林中任意域内创建与管理域用户账号。

15.2 添加用户账号

试图访问网络上资源的每个用户都必须具备一个用户账号, 且需要的账号类型依赖于网络配置。对活动目录域, 使用的是域用户账号; 对于工作组, 使用的是本地用户账号, 且只可以从属于特定的计算机。

15.2.1 创建域用户账号

创建域用户账号时, 要将用户的DN传递给DSADD USER命令。DN的常用名部分用于设置用户名, 其余部分用于指定用户账号在活动目录中的具体位置, 包括用户账号创建在哪个容器与相关域。比如, 要为cpandl.com域内Sales组织单元中的Lisa Andrews创建用户账号, 可以使用如下命令: **dsadd user "CN=Lisa Andrews,OU=Sales,DC=cpandl,DC=com"**。创建之后, 该账号将以Lisa Andrews作为用户登录名, 但由于没有指定其他属性, 因此, 出于安全原因, 该账号将被自动禁用。

用户名不区分大小写, 最长可以包括64个字符。典型情况下, 除了用户账号DN之外, 还需要指定如下一些要素。

- ☐ 首名, 以-Fn参数指定。
- ☐ 中间名, 以-Mi参数指定。
- ☐ 尾名, 以-Ln参数指定。
- ☐ 显示名, 以-Display参数指定。

注解 大多数情况下, 应该将显示名设置为与通常名一样的值, 用来对账号进行管理。只要知道了用户的显示名, 就可以获知用户DN中的通常名部分。

- SAM账号名（也被引用为登录名），以-Samid参数指定。
- 口令，以-Pwd参数指定。口令设置必须满足组策略中设定的复杂性需求。

注解 默认情况下，通常名中的前20个字符用于设置用户账号的SAM账号名，在Windows 2000以前也称为登录名。在域内，SAM账号名必须是唯一的，如果出现重叠，你可能需要将用户账号名设置为不同于显示名。在这种情况下，可能需要使用-Samid参数设置SAM账号名。

与使用Active Directory Users And Computers管理工具创建账号不同的是，这里不使用用户的首名、中间名与尾名等值来设置用户的显示名，而使用-Display参数来设置它。显示名是Windows在对话框中显示的。用户账号名的通常名部分与区分名的域名部分用于设置用户的完全限定登录名，用于用户的登录与认证。比如，如果用户的登录域为cpandl，登录名为lisaandrews，则完全限定登录名为cpandl\lisaandrews。

如果需要为使用这些参数的Lisa A. Andrews创建账号，可以使用如下命令：

```
dsadd user "CN=Lisa Andrews,OU=Sales,DC=cpandl,DC=com" -fn Lisa -mi A
-ln Andrews -samid "lisaandrews" -display "Lisa Andrews" -pwd dg56$2#
```

注解 注意上面命令中双引号的使用。在参数值中包含空格时，必须使用双引号封装该参数值。对用户DN、samid以及显示名等值，建议总是使用双引号对其进行封装。养成这样的习惯后，在需要双引号时就不会漏掉，从而保证命令的成功执行。否则就可能在需要的时候忘记使用，从而导致账号创建的失败。

如果创建账号时出现问题，则会看到警告信息，此时应该对语法进行检查，确保所有参数值正确设置，DN也是有效的。如果一切正常，DSADD USER会报告DSADD SUCCEEDED。

真实场景 在命令行中创建账号时，无论是用户账号还是组账号，最容易混淆的是很多不同的名称值。账号的通常名，也称为相对区分名，是指DN中使用第一个CN=指定的名组件，比如CN=Lisa Andrews。用户账号还包含一个显示名，在Windows对话框中使用。典型情况下，显示名是用户的完整名，你可能看到的是对完整名的引用而非显示名的引用。用户账号与组账号还包含一个pre-Windows 2000名。对用户，该名用于域登录与认证，因此也称为pre-Windows 2000登录名。

15.2.2 自定义域用户账号属性与组成员关系

所有新添加的域用户都是Domain Users组的成员，其主要属组也指定为Domain Users。你可以使用-Memberof参数添加组成员，其后跟随参数名与组的DN。如果组的DN中包含空格，则应该用双引号对其进行封装，比如：

```
dsadd user "CN=Lisa Andrews,OU=Sales,DC=cpandl,DC=com"
-memberof "CN=Backup Operators,CN=Builtin,DC=cpandl,DC=com" "CN=DHCP
Administrators,CN=Builtin,DC=cpandl,DC=com"
```

注解 特别要注意的是，组DN之间要用空格进行分隔。如果不使用空格分隔，则组成员关系将无法正确配置，并会返回错误信息。

上面的命令创建了用户账号，并将其添加为Backup Operators与DHCP Administrators组的成员。这是一个两步骤的过程。首先是添加账号，之后配置组成员关系。如果添加组成员时发生错误，则DSADD USER会说明，对象创建成功，但创建后出错。在指定组DN时，要检查其语法是否正确，之后使用DSMOD USER正确地配置组成员关系。

出于安全原因，创建账号时，可能需要设置如下一些参数。

- **-mustchpwd{yes|no}**。默认情况下，用户第一次登录时并不必须更改口令，即此时设置为**-mustchpwd no**。如果设置的是**-mustchpwd yes**，则用户第一次登录时必须更改口令。
- **-canchpwd{yes|no}**。默认情况下，用户可以更改自己的口令，即此时设置为**-canchpwd yes**。如果设置的是**-canchpwd no**，则用户无法更改自己的口令。
- **-pwdneverexpires{yes|no}**。默认情况下，设置为**-pwdneverexpires no**，即根据组策略，用户口令会在一定时间间隔后过期。如果设置的是**-pwdneverexpires yes**，则用户口令永不过期。

注解 使用**-pwdneverexpires yes**参数会重写域账号策略。通常，将账号口令设置为永不过期并不是一个好做法，这会使得口令的存在失去意义。

- **-disable{yes|no}**。默认情况下，创建用户账号与口令后，该账号就是可用的，即此时设置为**-disable no**。如果设置的是**-disable yes**，则该账号会被禁用。这种方法会临时性地防止任何人使用该账号。

要了解DSADD USER更多信息，参考如下实例。

为cpandl.com域内User容器中的Scott L. Bishop创建一个账号。为其设置口令，并且该口令在第一次登录时必须更改：

```
dsadd user "CN=Scott L.
Bishop,CN=Users,DC=cpandl,DC=com" -fn Scott -mi L -ln Bishop -samid
"scottb" -display "Scott L. Bishop" -pwd acornTree -mustchpwd yes
```

为ny.cpandl.com域内Engineering OU中的Bob Kelly创建一个账号。为其设置口令，该口令永不过期，但账号禁用：

```
dsadd user "CN=Bob
Kelly,OU=Engingeering,DC=ny,DC=cpandl,DC=com" -fn Bob -ln Kelly -
samid "bkelly" -display "Bob Kelly" -pwd dazedOne
-pwdneverexpires yes -disabled
```

为cpandl.com域内Marketing OU中的Eric F. Lang创建一个账号。为其设置口令，该口令不允许更改：

```
dsadd user "CN=Eric F.
Lang,OU=Marketing,DC=cpandl,DC=com" -fn Eric -mi F -ln Lang -samid
"eflang" -display "Eric F. Lang" -pwd albErt -canchpwd no
```

提示 你可以在域森林中任何具备访问权限的域内创建账号。有些情况下，你可能需要直接登录到某个域的域控制器。你可以使用-S *Server*连接到域森林内任意域内的特定域控制器，使用-D *Domain*连接到指定域内任意可用的域控制器。

大多数情况下，本节中讨论的这些参数在创建账号时已经足够。可以看出，基于DSADD USER语法，还有很多其他的用户账号参数，本章后面部分将讨论如何使用这些参数设置用户账号的属性。

15.2.3 创建本地用户账号

本地机器账号是在单独的计算机上创建的。如果需要为某个特定的计算机创建本地机器账号，必须本地登录该计算机，或者通过远程登录来打开本地的命令提示符。登录本地计算机后，你可以使用NET USER命令创建账号。有些情况下，本地计算机策略设置只允许使用待创建的账号名与/Add参数来创建账号，比如：

```
net user wrstanek /add
```

注解 不能在域控制器上创建本地用户账号，域控制器上不包含本地机器账号。

上面的命令中，创建了一个本地账号，该账号登录名为**wrstanek**，口令为空。尽管可以使用空口令，但这可能会使计算机与网络面临安全威胁。因此，对新创建的本地用户账号，建议同时提供用户名与口令，口令跟随在账号名之后，如下面实例所示：

```
net user wrstanek dg56$2# /add
```

上面的命令创建了本地机器账号**wrstanek**，口令为**dg56\$2#**。

如果成功创建了账号，则NET USER会声明“命令成功完成”。然而，如果在账号创建过程中遇到其他问题，NET USER并不会显示错误消息本身，而是显示命令语法。这种情况下，就需要检查语法，并确保相关参数值的设置是正确的。

对本地用户账号，可能使用的其他值与参数包括下面列举的。

- **/comment: "DescriptionText"**。为用户账号设置描述信息。通常，设置的是用户的工作头衔或部门。
- **/fullname: "Name"**。为用户账号设置完整名，完整名也被引用为显示名。
- **/passwordchg{yes|no}**默认情况下，用户可以更改自己的口令，即此时设置为 **/passwordchg yes**。如果设置的是 **/passwordchg no**，则用户无法更改自己的口令。
- **/passwordreq{yes|no}**。默认情况下，用户账号需要设置相应口令，即此时设置为 **/passwordreq yes**。因此，用户账号必须设置相应口令，而不能为空。
- **/active{yes|no}**。默认情况下，用户账号在创建后是激活的，即此时设置为 **/active yes**。如果设置的是 **/active no**，则用户账号创建后将被禁用。这一参数可用于临时阻止任何人使用账号。要了解使用NET USER的更多信息，参考如下实例。

使用完整名与描述信息，为Desktop Support组创建一个本地机器账号：

```
net user dsupport squ5 /fullname:"Desktop Support"
/comment:"Desktop Support Account" /add
```

为Phil Spencer创建一个本地机器账号，包含完整名与描述信息，并要求使用口令：

```
net user pspencer magma2 /fullname:"Phil Spencer"
/comment:"Offsite Sales Manager" /passwordreq yes /add
```

为Chris Preston创建一个本地机器账号，包含完整名与描述信息，设置口令，但不允许用户更改口令：

```
net user chrisp apples /fullname:"Chris Preston" /comment:"PR
Manager" /passwordchg no /add
```

15.3 管理用户账号

与使用活动目录用户和计算机管理工具相比，从命令行管理账号可以使用更多的选项，尤其是在同时操作多个账号时容易得多。

15.3.1 查看与查找用户账号

你可以使用DSQUERY USER命令查找用户。你可以根据通常名、SAM账号名、描述信息等搜索账号，也可以使用这些字段的通配符进行查找。DSQUERY USER的输出包含了与搜索标准匹配的用户DN，并且输出信息可以通过管道用作其他命令的输入，包括DSGET USER，并用于显示账号的相关属性。

DSQUERY USER与DSGET USER最适合于一起使用。先使用DSQUERY USER获取一个或多个用户的DN，之后使用DSGET USER显示相关账号的属性，这些属性是通过不同参数指定的，包括下面列举的。

- -display。显示输出信息中匹配用户账号的完整名属性。
- -desc。显示输出信息中匹配用户账号的描述信息。
- -dn。显示输出信息中匹配用户账号的区分名。
- -empid。显示输出信息中匹配用户账号的雇员ID属性。
- -fn。显示输出信息中匹配用户账号的首名属性。
- -mi。显示输出信息中匹配用户账号的中间名属性。
- -ln。显示输出信息中匹配用户账号的尾名属性。
- -samid。显示输出信息中匹配用户账号的SAM账号名。
- -sid。显示输出信息中匹配用户账号的安全标识符。
- -disabled。显示“是|否”，表示用户账号是否禁用。
- -effectivepsa。显示输出信息中匹配用户账号的有效的口令设置对象（PSO）。

注解 活动目录中定义了3种账号策略：口令策略、账号锁定策略以及Kerberos策略。PSO用于定义域中账号策略设置，域运行在Windows Server 2008功能层时，PSO是可用的。

DSGET USER以表格形式显示输出信息。一般来说，你应该总是使用-Dn、-Samid、-Display等参数，以便清晰地区分与标示输出信息中的用户账号。比如，如果需要搜索所有禁用的engineering用户，可以使用如下命令：

```
dsquery user "OU=Eng,DC=cpandl,DC=com" | dsget user -dn -disabled
```

上面的命令将列出cpandl.com域内Engineering OU中所有处于禁用状态的用户账号，比如：

dn	disabled
CN=edwardh,OU=Eng,DC=cpandl,DC=com	yes


```

CN=jacob1,OU=Eng,DC=cpand1,DC=com      yes
CN=maryk,OU=Eng,DC=cpand1,DC=com        yes
CN=ellene,OU=Eng,DC=cpand1,DC=com        yes
CN=williams,OU=Eng,DC=cpand1,DC=com      yes
dsget succeeded

```

你可以使用如下命令显示SAM账号名:

```

dsquery user -name william* | dsget user -samid -disabled
samid      disabled
williamb    yes
williamd    yes
williams    no
dsget succeeded

```

这一命令搜索所有通常名以William引导的用户账号,并显示每个匹配用户账号的SAM账号名与禁用状态。

15.3.2 确定单独用户账号的组成员关系

你可以使用DSGET USER的另一种语法格式来获取单独用户账号的组成员关系。比如,如果需要查看WilliamS属于哪些组,可以使用如下命令:

```
dsquery user -name williams | dsget user -memberof
```

或

```
dsget user "CN=William Stanek,OU=Eng,DC=cpand1,DC=com" -memberof
```

上面两条命令的工作方式是一样的。第一条命令中,使用DSQUERY USER获取用户账号的DN。第二条命令则直接指定DN。两种方式都将展示类似于如下的组成员关系信息:

```

"CN=Tech,CN=Users,DC=cpand1,DC=com"
"CN=Engineering,OU=Eng,DC=cpand1,DC=com"
"CN=Domain Users,CN=Users,DC=cpand1,DC=com"

```

输出信息表明,用户是Tech、Engineering、Domain Users等组的成员。

需要指出的是,尽管可以使用上面的技术显示多个用户的组成员关系,但不能显示相关联用户的DN或SAM账号名。因而,输出信息中用于区分组成员关系针对的是哪一个用户的唯一标识是空白行,比如,使用如下查询命令:

```
dsquery user -name bill* | dsget user -memberof
```

得到的输出信息将类似于如下格式:

```

"CN=Tech,CN=Users,DC=cpand1,DC=com"
"CN=Engineering,OU=Eng,DC=cpand1,DC=com"
"CN=Domain Users,CN=Users,DC=cpand1,DC=com"

"CN=Domain Users,CN=Users,DC=cpand1,DC=com"

"CN=Tech,CN=Users,DC=cpand1,DC=com"
"CN=Engineering,OU=Eng,DC=cpand1,DC=com"
"CN=Domain Users,CN=Users,DC=cpand1,DC=com"

"CN=Engineering,OU=Eng,DC=cpand1,DC=com"

```



```
"CN=Domain Users,CN=Users,DC=cpand1,DC=com"
```

```
"CN=Tech,CN=Users,DC=cpand1,DC=com"
```

```
"CN=Engineering,OU=Eng,DC=cpand1,DC=com"
```

```
"CN=Domain Users,CN=Users,DC=cpand1,DC=com"
```

```
"CN=Domain Users,CN=Users,DC=cpand1,DC=com"
```

```
"CN=Domain Users,CN=Users,DC=cpand1,DC=com"
```

上面的输出信息分别显示了7个用户账号的组成员关系，之所以这样说是根据其中包含的空白行，但是你无法确定具体针对的是哪7个用户账号。

15.3.3 设置或更改用户账号属性

从命令行中，使用DSMOD USER命令设置或更改用户账号属性是快捷容易的。你可以为单独的用户设置或更改属性，也可以同时为多个用户设置或更改属性。比如，如果需要将Sales OU中150个用户账号的部门属性设置为Sales & Marketing，公司属性设置为City Power and Light，头衔设置为Customer Sales，你可以使用如下命令完成这些设置任务：

```
dsquery user "OU=Sales,DC=cpand1,DC=com" | dsmod user -dept "Sales  
& Marketing" -company "City Power and Light" -title "Customer Sales"
```

执行之后，DSMOD USER将报告每一更改行为的成功或失败：

```
dsmod succeeded:CN=edwardh,OU=Sales,DC=cpand1,DC=com no  
dsmod succeeded:CN=erinp,OU=Sales,DC=cpand1,DC=com no  
dsmod succeeded:CN=jayo,OU=Sales,DC=cpand1,DC=com no  
dsmod succeeded:CN=johng,OU=Sales,DC=cpand1,DC=com yes  
...  
dsmod succeeded:CN=williams,OU=Sales,DC=cpand1,DC=com yes
```

在GUI工具中，完成这些修改可能会需要几个小时的时间，而在命令行中则只需要几分钟的时间。你只需要键入相应的命令，余下的工作将由DSMOD USER完成。

其他可能需要经常使用的参数包括下面4个。

- ❑ -webpg。为用户设置intranet或Internet地址，地址将会在目录列表中显示，比如\\Intranet\Sales。
- ❑ -profile。为用户的profile（设置了用户账号的环境信息）设置路径，比如\\Gamma\Profiles\wrstaneck。
- ❑ -hmdrv。为用户宿主目录设置驱动器盘符，比如X:。用户宿主目录将映射到这一盘符。
- ❑ -hmdir。为用户设置宿主目录，比如\\Gamma\Users\wrstaneck。

警告 通常，在用户登录时，不能修改用户配置文件路径、宿主驱动器或宿主目录，因为这可能会出现问題。如果需要更新这些信息，可以在正常工作时间之外（或者需要用户暂时退出登录几分钟）改变这些信息。

提示 默认情况下，进行上述更新时如果出错，则DSMOD USER会报告出错并停止执行。通常，这是期望的行为，因为错误的更改是不需要的。但你也可以使用-C参数，使DSMOD USER在报告出错后继续执行。

这些参数可以接受特殊值`$username$`。通过这一值，可以为单独的用户名指定路径与文件名。比如，如果将宿主目录路径指定为`\\Gamma\Users$username$`或`C:\Home$username$`，则Windows会使用实际的用户名来替换`$username$`。比如，如果当前操作的用户为`erinb`、`sandyr`、`miked`与`kyler`，则Windows将分别为这些用户指定单独的宿主目录`\\Gamma\Users\erinb`、`\\Gamma\Users\sandyr`、`\\Gamma\Users\miked`、`\\Gamma\Users\kyler`，或者`C:\Home\erinb`、`C:\Home\sandyr`、`C:\Home\miked`、`C:\Home\kyler`。其中，`\\Gamma\Users\`是网络共享路径，`C:\Home\`则代表用户计算机中的目录路径。

根据上面的讲述，你可以为Sales OU中的所有用户设置Web page、profile、宿主驱动器以及宿主目录等内容，使用如下命令：

```
dsquery user "OU=Sales,DC=cpandl,DC=com" | dsmod user -webpg
\\Intranet\Sales$username$ -profile "\\corpd02\sales$username$"
-hmdrv "X:" -hmdir "\\corpserver01\users$username"
```

真实场景

在活动目录用户和计算机管理工具中，输入值`%username%`，可以获取路径与文件名（以单独用户为基础）。不要将这里讨论的特殊参数与该值一起使用。`%username%`是一个环境变量，该GUI工具会以每个用户为基础对该环境变量进行相应替换。然而，在命令行中，会根据当前登录用户来对这一值与其他环境变量进行解释。因此，在这一实例中，`%username%`的取值将是运行该命令的用户账号。

15.3.4 禁用与激活用户账号

在命令行中，使用DSMOD USER命令与-Disabled参数，可以激活或禁用用户账号。使用-**disabled yes**参数，可以禁用用户账号；使用-**disabled no**参数，可以激活用户账号。

下面的实例中，禁用OffsiteUsers OU中的所有用户：

```
dsquery user "OU=OffsiteUsers,DC=cpandl,DC=com" | dsmod user -disabled yes
```

运行之后，DSMOD USER会报告每一更改的成功或失败。

15.3.5 重置过期的用户账号

你可以为域用户账号设置一个特定的到期日期，并使用DSGET USER与-Acctexpires参数来检查账号的到期日期。比如，如果需要检查Sales OU中所有用户账号的到期日期，可以使用如下命令：

```
dsquery user "OU=Sales,DC=cpandl,DC=com" | dsget user -dn -acctexpires
```

输出信息包含了Sales OU中每一用户账号的到期日期，并根据账号的区分名列出，比如：

```
dn      acctexpires
CN=Lisa Andrews,OU=Sales,DC=cpandl,DC=com    never
CN=Joseph Brad,OU=Sales,DC=cpandl,DC=com      11/15/2010
CN=Ann Beebe,OU=Sales,DC=cpandl,DC=com        never
CN=Jeanne Bosworth,OU=Sales,DC=cpandl,DC=com  12/31/2010
dsget succeeded
```

这里，没有到期日期的账号的这一数值设置为“永不过期”，其他账号则有一个具体的到期日期，比如11/15/2010。

如果需要扩展或改变账号的到期日期，以使用户登录某个域，可以使用DSMOD USER命令，并将

-Acctexpires参数设置为需要该账号有效的天数。比如，如果某账号在下一个60天内应该有效，则可以键入-acctexpires 60，比如：

```
dsquery user -name johnw | dsmod user -acctexpires 60
```

或

```
dsmod user "CN=John Woods,OU=Sales,DC=cpandl,DC=com" -acctexpires 60
```

上面的命令中，改变了John Woods的到期日期。

如果需要移除账号的到期日期，可以使用0，表示该账号永不过期，比如：

```
dsquery user -name johnw | dsmod user -acctexpires 0
```

注解 要设置一个账号使其已经过期，可以使用负数，比如-acctexpires -1。

15.3.6 控制与重置用户口令

使用DSGET USER命令，可以检查用户账号的口令设置。典型情况下，你可能需要知道某用户是否可以改变口令、口令是否到期、口令加密机制是否可逆等，你可以分别使用-Canchpwd、-Pwdneverexpires、-Reversiblepwd等参数来检查这些设置。你也可能需要知道账号是否设置为用户下次登录时必须改变口令，为此，可以使用-Mustchpwd参数。比如，如果需要检查Users容器中所有用户的这些设置，可以使用如下命令：

```
dsquery user "CN=Users,DC=cpandl,DC=com" | dsget user -samid -canchpwd  
-pwdneverexpires -reversiblepwd -mustchpwd
```

执行之后，输出信息将根据SAM账号名展示Users容器所有用户的相关口令设置，比如：

samid	mustchpwd	canchpwd	reversiblepwd	pwdneverexpires
andya	no	yes	no	no
billg	no	yes	no	no
bobh	yes	yes	no	no
brianw	no	yes	no	no
conniej	no	yes	yes	yes

dsget succeeded

DSMOD USER还提供了几个参数，用于控制这些以及其他的口令设置。你可以使用-Pwd参数为特定用户账号设置口令，之后按下面列举的使用方式配置口令。

- 使用-mustchpwd yes参数，该参数的作用是强制用户在下次登录时修改口令。
- 使用-canchpwd no参数，该参数的作用是使得用户无法修改口令。
- 使用-pwdneverexpires yes参数，该参数的作用是使得账号口令永不过期，这将覆盖组策略中的设置。

命令行的精彩之处在于，可以同时控制与修改多个用户账号的口令。假定需要将TempEmployee OU中每个用户的口令设置为Time2ChangeMe，并强制这些用户下次登录时修改口令，可以使用如下命令：

```
dsquery user "OU=TempEmployee,DC=cpandl,DC=com" | dsmod user -pwd  
Time2ChangeMe -mustchpwd yes
```

15.3.7 移动用户账号

通常，用户账号放置在Users容器或OU中。使用DSMOVE命令，可以将用户账号移动到同一个域内的其他容器或OU中。指定用户账号的当前DN，之后使用-Newparent参数指定该用户账号需要移动到的新位置或父DN。比如，如果需要将用户账号William Stanek从Tech OU移动到Engineering OU，则应该指定用户账号DN，比如“CN=William Stanek,OU=Tech,DC=cpandl,DC=com”。之后指定新位置的父DN，比如“OU=Engineering,DC=cpandl,DC=com”，相应命令类似于如下：

```
dsmove "CN=William Stanek,OU=Tech,DC=cpandl,DC=com" -newparent
"OU=Engineering,DC=cpandl,DC=com"
```

在执行DSMOVE命令之前，你可能需要使用DSQUERY USER获取用户账号DN，之后将DSQUERY USER的输出通过管道传递给DSMOVE，参考如下实例：

```
dsquery user -name "William Stanek" | dsmove
-newparent "OU=Engineering,DC=cpandl,DC=com"
```

这里，用户账号DN，“CN=William Stanek,OU=Tech,DC=cpandl,DC=com”是通过DSQUERY USER获取的，并用作DSMOVE的输入。

15.3.8 用户账号重命名

尽管移动用户是直截了当的，但对用户账号进行重命名则需要做一些工作。对用户账号进行重命名时，为账号赋予了新的通常名。在用户结婚、离婚或过继时，可能都需要为其进行重命名。比如，如果Nancy Anderson (nancya) 结婚了，她可能就需要将用户名改变为Nancy Freehafer (nancyf)。对其账号进行重命名时，所有相关联的特权与访问权限都将反映账号名的改变。因而，如果查看nancya具有访问权限的文件许可权限，会发现nancyf具有访问权限（而nancya将不再列出）。

你可以使用DSMOVE命令对账号进行重命名。指定用户DN，之后使用-Newname参数指定新的通常名。比如，如果需要将用户对象Nancy Anderson重命名为Nancy Freehafer，则可以使用如下命令：

```
dsmove "CN=Nancy Anderson,OU=Marketing,DC=cpandl,DC=com"
-newname "Nancy Freehafer"
```

你也可以使用DSQUERY USER获取用户DN，参考如下实例：

```
dsquery user -name N*Anderson | dsmove -newname "Nancy Freehafer"
```

上面命令中，DSQUERY USER查找以字母N引导，以Anderson结束的账号，之后使用DSMOVE对满足这一标准的账号进行重命名。

对用户账号进行重命名并不会改变其他的账号属性。由于有些属性会反映旧的用户名，因此需要使用DSMOD USER更新这些属性，以便反映更改之后的账号名。需要修改的参数包括下面6个。

- -Ln。用于改变用户账号的尾名。
- -Display。用于改变用户账号的显示名。
- -Samid。用于改变用户账号的SAM账号名。
- -Profile。用于改变用户账号的配置文件路径。之后，需要对磁盘上相应目录进行重命名。
- -Loscr。如果为每个用户使用单独的登录脚本，则可以使用-Loscr改变登录脚本名属性。之后，需要对磁盘上相应的登录脚本进行重命名。
- -Hmdir。用于改变宿主目录路径。之后，需要对磁盘上相应的目录进行重命名。

注解 大多数情况下，在用户登录时，不能修改这一信息，因为这可能会出现问題。如果需要更新这些信息，可以在正常工作时间之外（或者需要用户退出登录几分钟）改变这些信息。

参考如下实例：

```
dsquery user -name N*Freehafer | dsmod -samid nancyf -ln Freehafer
-display "Nancy Freehafer"
```

上面的命令中，对SAM账号名、尾名以及显示名进行了修改，以便配合前面对用户Nancy Freehafer所做的重命名。

真实场景 用户名用于更方便地对用户账号进行管理与使用。在更深的系统层面，实际上，Windows Server 2008使用账号的安全标识符（SID）来识别、追踪与处理用户账号，而不是使用用户名。SID是唯一的标识符，在创建账号时由系统生成。由于在系统内部，SID与账号名存在一个映射关系，因此不需要改变重命名账号的特权或许可权限，Windows Server 2008会在必要的时候将SID映射到新账号名。

15.3.9 删除用户账号

如果不再需要一个用户账号，可以使用DSRM命令将其从活动目录中永久性删除。大多数情况下，你可能需要删除某个特定的计算机账号，比如Lisa Andrews。如果是这种情况，你可以将该计算机账号的DN传递给DSRM命令，比如：

```
dsrm "CN=Lisa Andrews,OU=Sales,DC=cpandl,DC=com"
```

默认情况下，DSRM会弹出提示信息询问是否确认删除。如果不希望看到提示信息，可以使用-Noprompt参数，比如：

```
dsrm "CN=Lisa Andrews,OU=Sales,DC=cpandl,DC=com" -noprompt
```

注解 即便删除了用户账号，Windows Server 2008并不会删除该用户的配置文件、个人文件或宿主目录。如果需要删除这些文件与目录，就必须手工删除。如果这是一个需要常规性执行的任务，可以创建一个脚本来完成。要记住的是，在进行这些操作之前，应该对文件或数据进行备份。

15.4 从命令行管理组账号概览

组账号用于管理多个用户的特权，Windows Server 2008包含3种类型的组账号。

- **安全组**。带有安全描述符的组，用于管理访问许可权限。你可以使用目录服务命令创建与管理安全组。
 - **分发组**。用作电子邮件分发列表的组，不带有安全描述符。你可以使用目录服务命令创建与管理分发组。
 - **本地组**。只在本地计算机上使用的组，你可以使用网络服务命令创建与管理本地组。
- 安全组与分发组在域内使用，在整个目录内都是可用的；本地组则只在其创建机器上可用。通常

的域组账号命令行工具包括下面3个。

- **DSADD GROUP**。在活动目录中创建组账号，其语法格式如下：

```
dsadd group GroupDN [-secgrp {yes | no}] [-scope {l | g | u}]
[-samid SAMName] [-desc Description] [-memberof Group ...]
[-members Member ...] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-q] [{-uc | -uco | -uci}]
```

- **DSGET GROUP**。显示组账号的属性，有两种语法。

显示多个组账号的语法格式如下：

```
dsget group GroupDN ... [-dn] [-samid] [-sid] [-desc] [-secgrp]
[-scope] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
[-part PartitionDN [-qlimit] [-qused]]
```

查看单独组的组成员关系信息的语法格式如下：

```
dsget group GroupDN [{-memberof | -members} [-expand]] [
{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-c]
[-q] [-l] [{-uc | -uco | -uci}]
```

- **DSMOD GROUP**。在活动目录中修改一个或多个组账号的属性，其语法格式如下：

```
dsmod group GroupDN ... [-samid SAMName] [-desc Description]
[-secgrp {yes | no}] [-scope {l | g | u}] [{-addmbr | -rmmbr |
-chmbr} MemberDN ...] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-c] [-q] [{-uc | -uco | -uci}]
```

提示 这些命令可以接受来自DSQUERY GROUP的输入信息，用来为待操作的安全组设置区分名(DN)。你也可以为每个组键入DN，但要确保每个DN之间使用空格分隔。

要对本地组账号进行管理，可以使用NET LOCALGROUP命令。该命令具有几种语法格式，具体语法格式依赖于要完成的任务，如下所示。

- **创建本地组账号**：net localgroup [GroupName {/add[/comment: "Text"]}]。
- **修改本地组账号**：net localgroup [GroupName Name...]{/add | /delete}。
- **删除本地组账号**：net localgroup [GroupName {/delete[/comment: "Text"]}]。

注解 NET LOCALGROUP用于将一个本地组添加到当前(登录)域内的其他组。在有些受限的环境下，你可能需要这样做，但一般不要使用这一技术为常规用户赋予更高的访问许可权限。比如，如果创建了一个名为DevTesters的本地组，并将其添加到Developers域组中，则作为DevTesters组成员的本地机器用户也将具备与Developers域组成员一样的域许可权限。这样做的原因在于，正在测试本地系统配置的开发人员需要对域的访问权限。

15.5 添加组账号

需要使用的组账号类型依赖于网络配置。在域中，典型情况下使用的是安全组与分发组；在工作组中，典型情况下使用的是本地组，且只可以从属于特定的计算机。

15.5.1 创建安全组与分发组

前面讲述了，安全组用于管理组用户的访问许可权限，分发组用于电子邮件分发列表。不管创建的是安全组还是分发组，该组的使用方法都依赖于范围。这里，范围是指该组有效地作用区域。定义的范围包括下面列举的3个。

- **域本地组**。用于在单一域内授权的组，该组的成员可以只包括其定义域内的账号（用户账号与计算机账号）与组。
- **全局组**。用于在域树或域森林内对任意域中对象进行授权的组，该组的成员可以只包括其定义域内的账号与组。
- **通用组**。用于在整个域树或域森林的更广范围内进行授权的组，该组成员可以包括账号、全局组以及域树或域森林中任意域内的其他通用组。

注解 只有当活动目录运行在Windows 2000本原功能层或更高时，通用安全组才是可用的，并且在大型网络上比小型网络上具有更强大的功能。这主要是因为为管理员增加了另外的组层级，因而，在大型网络（需要对组进行更多的控制）中优势比较明显。

基于范围与操作模式，表15-1总结了组能力。从表中可以看出，二者都会对组的操作产生影响。

表15-1 组能力及其功能层与范围

组 能 力	域本地组	全 局 组	通 用 组
Windows 2000本原功能层或更高	成员可以包括任意域中的用户账号、全局组以及通用组，或同一域内的域本地组	成员只可以包括同一域内的用户账号与全局组	成员可以包括任意域内的用户账号，以及任意域内（不管范围）的组
Windows 2000混合功能层	成员可以包括任意域内的用户账号与全局组	成员只可以包括同一域内的用户与组账号	在混合模式域内不能创建通用安全组
成员关系	可以放置到其他域本地组，只可以在同一域内赋予许可权限	可以放置到其他组，并可以在任意域内赋予许可权限	可以放置到其他组，并可以在任意域内赋予许可权限

创建组时，要将组DN传递给DSADD GROUP命令。DN的常用名部分用于设置组的显示名，其余部分用于指定组在活动目录中的具体位置，包括组创建在哪个容器与相关域。默认情况下，如果没有提供其他参数，则会创建一个全局安全组。比如，要在cpandl.com域内Sales组织单元中创建一个名为Sales的全局安全组，可以使用如下命令：**dsadd group "CN=Sales,OU=Sales,DC=cpandl,DC=com"**。创建之后，将以Sales作为组的显示名与SAM账号名，但没有设置其他属性。

组名不区分大小写，最长可以包括64个字符。典型情况下，可以直接指定组类型与范围。你可以使用-Secgrp参数指定某个组是否为安全组，如下所示。

- **-secgrp yes**。表示正在创建的是安全组。
 - **-secgrp no**。表示正在创建的是分发组。
- 要设置组范围，可以使用-Scope参数，如下所示。
- **-scope l**。表示创建一个本地域组。
 - **-scope g**。表示创建一个全局组。

- **-scope u**。表示创建一个通用组。对于安全组，只有当活动目录运行在Windows 2000本原功能层或更高时，通用组才是有效的。

注解 默认情况下，组被创建为安全组，作用范围为全局范围。因此，即使使用不同的范围创建安全组，也不需要指定**-secgrp yes**参数，因为这是默认的。

组名中的前20个字符用于设置组的SAM账号名——在Windows 2000以前也称为组名。在域内，SAM账号名必须是唯一的，如果出现重叠，你可能需要将组的SAM账号名设置为不同于显示名。在这种情况下，可能需要使用**-samid**参数设置SAM账号名。

创建组时，还可以指定组成员关系。如果当前创建的组应该是现存组的成员，则可以使用**-Memberof**参数指定这些组的DN。如果该组应该包含一些用户或其他组作为自己的组成员，则可以使用**-Members**参数指定这些成员的DN。然而，使用DSMOD GROUP配置组成员关系会更加容易，因为可以将DSQUERY USER的输出信息中DN列表作为DSMOD GROUP的输入，从而不再需要键入数十个甚至数百个DN。

要了解如何创建组，参考如下一些实例。

创建一个名为Engineering的域本地安全组，并将该组添加到tech.cpandl.com域内的Engineering OU：

```
dsadd group "CN=Engineering,OU=Engineering,DC=tech,DC=cpandl,DC=com" -scope l
```

在cpandl.com域内的Users容器中创建一个名为Engineering Global的全局安全组，并将其SAM账号名设置为gEngineering：

```
dsadd group "CN=Engineering Global,CN=Users,DC=cpandl,DC=com" -samid "gEngineering"
```

在cpandl.com域内的Engineering OU中创建一个名为Engineering All的通用分发组的全局安全组，并将其SAM账号名设置为allEngineering：

```
dsadd group "CN=Engineering All,OU=Engineering,DC=cpandl,DC=com" -samid "allEngineering" -secgrp no -scope u
```

如果在创建组时遇到问题，会看到一条警告消息。此时需要检查命令语法，确保所有参数值正确设置，并且DN是有效的。如果一切正常，则DSADD GROUP会报告DSADD SUCCEEDED。组创建之后，你可以为其添加成员，并设置其他参数，本章后面将进行讨论。

15.5.2 创建本地组并为其分配成员

本地组是在单独的计算机上创建的，其作用是对本地登录（而非登录到域）的用户进行许可权限管理。要创建一个本地组，需要登录到待操作的计算机，或者通过远程登录获取一个本地命令提示符。登录之后，可以使用NET LOCALGROUP创建本地组账号。

键入NET LOCALGROUP命令，其后跟随组名，之后使用/Add参数，参考如下实例：

```
net localgroup localDevs /add
```

注解 不能在域控制器上创建本地组账号，域控制器上不包含本地机器账号。

上面的命令中，在本地计算机上创建了名为localDevs的组。如果需要，你也可以使用/Comment参数为该组指定描述信息，比如：

```
net localgroup localDevs /comment:"Local Developers and Testers" /add
```

如果成功创建了账号，则NET LOCALGROUP会声明“命令成功完成”。然而，如果在账号创建过程中遇到其他问题，NET LOCALGROUP并不会显示错误消息本身，而是显示命令语法。这种情况下，就需要检查语法，并确保相关参数值的设置是正确的。

创建本地组时，也可以指定本地用户账号列表，并将这些账号作为该组的成员，账号列表跟随在组名之后，如下所示：

```
net localgroup localDevs williams johng edwardh /add
```

上面的命令中，创建了一个名为localDevs的组，并将WilliamS、JohnG、EdwardH添加为该组的成员。

如果需要在以后向本地组中添加成员，而不是在创建组时添加，其语法与创建组时一样。比如，下面的命令创建了一个名为custSupport的组：

```
net localgroup custSupport /add
```

如果以后需要为该组添加成员，则可以使用如下命令：

```
net localgroup custSupport williams johng edwardh /add
```

上面命令中，将WilliamS、JohnG、EdwardH添加为该组的成员。

15.6 管理组账号

与使用活动目录组和计算机管理工具相比，从命令行管理组账号可以使用更多的选项，尤其是在同时操作多个组账号时容易得多。

15.6.1 查看与寻找组账号

需要获取关于组账号的信息时，你可以使用DSQUERY GROUP命令。你可以根据通常名、SAM账号名、描述信息等进行搜索，也可以使用这些字段的通配符进行查找。DSQUERY GROUP的输出包含了与搜索标准匹配的组的DN，并且输出信息可以通过管道用作其他命令的输入，包括DSGET GROUP。

典型情况下，可以将DSQUERY GROUP与DSGET GROUP一起使用。先使用DSQUERY GROUP获取一个或多个组的DN，之后使用DSGET GROUP显示相关账号的属性。有用的DSGET GROUP参数包括下面列举的。

- -Desc。显示输出信息中匹配组账号的描述信息。
- -Dn。显示输出信息中匹配组账号的区分名。
- -Samid。显示输出信息中匹配组账号的SAM账号名。
- -Scope。显示输出信息中匹配组账号的范围，包括域本地、全局或通用。
- -Secgrp。如果该组是安全组，则显示yes。如果是分发组，显示no。

□ -Sid。显示输出信息中匹配用户账号的安全标识符。

与其他DSGET命令类似，DSGET GROUP以表格形式显示输出信息。一般来说，你应该使用-Dn、-Samid等参数，以便清晰地区分与标示输出信息中的组账号。比如，如果需要搜索所有可用的marketing组，可以使用如下命令：

```
dsquery group -name marketing* | dsget group -dn -scope -secgrp
```

上面命令的执行结果将显示DN、范围以及安全组信息：

```
dn      scope      secgrp
CN=MarketingAll,OU=Sales,DC=cpandl,DC=com    universal    no
CN=Marketing Global,OU=Sales,DC=cpandl,DC=com  global      no
CN=Marketing Local,OU=Sales,DC=cpandl,DC=com   domain local no
dsget succeeded
```

15.6.2 确定组成员关系

如果需要确定组成员关系，你可以使用DSGET GROUP的另一种语法格式，这需要两个特殊的参数：-Members与-Memberof。-Members参数可以用于确定哪些用户与组属于某个特定组，-Memberof参数用于确定指定的组属于哪一个组。要了解这些参数如何工作，假定需要查看名为AllUsers的组的当前成员，可以使用如下命令：

```
dsquery group -name AllUsers | dsget group -members
```

或者你也可以直接键入组DN，比如：

```
dsget group "CN=AllUsers,CN=Users,DC=cpandl,DC=com" -members
```

上面命令中，组存在于cpandl.com域内的Users容器中。无论哪条命令，输出信息中都会展示该组成员的DN，比如：

```
"CN=Tech,OU=Tech,DC=cpandl,DC=com"
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Sales,OU=Sales,DC=cpandl,DC=com"
"CN=Domain Users,CN=Users,DC=cpandl,DC=com"
```

从输出信息中可以看出，AllUsers组包括Tech、Engineering、Sales、Domain Users组等成员。AllUsers组也可能包含用户账号作为其成员。

如果需要确定某个组属于哪一个组，可以使用-Memberof参数。比如，DevUsers组为Domain Administrators组与Developers组的成员，通过如下命令，可以显示这种成员关系信息：

```
dsquery group -name devusers | dsget group -memberof
```

或

```
dsget group "CN=devusers,OU=Dev,DC=cpandl,DC=com" -memberof
```

上面两条命令的工作方式是一样的。第一条命令中，使用DSQUERY GROUP获取组账号的DN。第二条命令中，直接指定DN。两条命令的输出信息都会包含组列表，DevUsers组是这些组的成员。

注解 你可以使用上面的两种技术显示多个组的成员关系信息。然而，你无法显示相关组的DN或SAM账号，因为DSGET GROUP的第二种语法不支持这一点。

15.6.3 改变组类型或范围

创建组之后，有时可能需要改变组类型或范围。这并不特别容易，因为有很多控制因素用于防止意外的改变，以免影响整个组织内的访问权限。首先，在Windows 2000混合功能层或Windows Server 2003 Interim功能层，不能改变组类型与范围。在Windows 2000 本原、Windows Server 2003或Windows Server 2003功能层，有如下一些约束。

- **域本地组**。可以转换为通用组，前提是该组不包含同时也是其他具有域本地组成员的成员。
- **全局组**。可以转换为通用组，前提是该组不是任意其他全局组的成员。
- **通用组**。可以转换为任意其他组。要记住的是，全局组不能包含一个通用组作为其成员，且本地组只能是其他本地组的成员。

根据这些约束，可以使用DSMOD GROUP与-Secgrp参数来改变组类型，如下所示。

- 使用-secgrp yes，将分发组改变为安全组。
- 使用-secgrp no，将安全组改变为分发组。

参考如下一些实例。

将Engineering安全组转换为分发组：

```
dsquery group -name Engineering | dsmod group -secgrpno
```

将AllMarketing分发组转换为安全组：

```
dsmod group "CN=AllMarketing,OU=Marketing,DC=cpandl,DC=com"
-secgrp yes
```

你可以使用DSMOD GROUP的-Scope参数改变组范围，如下所示。

- 使用-scope l，将范围设置为域本地。
- 使用-scope g，将范围设置为全局。
- 使用-scope u，将范围设置为通用。

参考如下实例。

将Marketing组的范围设置为域本地：

```
dsquery group -name Marketing | dsmod group -scope l
```

将Sales组的范围设置为全局：

```
dsmod group "CN=Sales,CN=Users,DC=cpandl,DC=com" -scope g
```

15.6.4 添加、移除或替换组成员

在命令行中，可以很容易地改变任意组的组成员关系。与GUI中类似，你可以很容易地添加或移除用户、组、计算机等组成员。不过，比GUI更强大的是，命令行工具可以同时添加或移除多个组成员。你也可以完全替换现有的组成员关系列表。

1. 向组内添加成员

比如，你可以使用单一的命令将Sales组织单元内的所有100个用户添加到AllSales组。为此，你可以使用DSQUERY USER命令获取待操作的用户账号列表，之后将其作为输入信息传递给DSMOD

GROUP。用于添加组成员的参数为-Addmbr，命令类似于如下格式：

```
dsquery user "OU=Sales,DC=ny,DC=cpandl,DC=com" | dsmod group "CN=AllSales,
OU=Sales,DC=ny,DC=cpandl,DC=com" -addmbr
```

上面的命令中，首先获取ny.cpandl.com域内Sales OU中所有用户账号的列表，之后将其作为输出传递给DSMOD GROUP。DSMOD GROUP将这些用户账号添加为AllSales组的成员，该组在ny.cpandl.com域内的Sales容器中。

使用-Addmbr的另一种方法是指定待添加对象的DN。比如，如果需要将SalesLocal组与SalesGlobal组添加到AllSales组中，可以使用如下命令：

```
dsquery group -name AllSales | dsmod group
-addmbr "CN=SalesLocal,OU=Sales,DC=ny,DC=cpandl,DC=com"
"CN=SalesGlobal,OU=Sales,DC=ny,DC=cpandl,DC=com"
```

注解 对象DN可以包含用户账号、组账号以及计算机账号。

2. 从组中移除成员

与-Addmbr参数相反的是-Rmmbr参数，该参数用于从组中移除成员。与-Addmbr参数类似，-Rmmbr参数可以从输入信息或空格分隔的列表中接受对象DN。因此，如果需要从AllSales组中移除所有市场与客户支持用户，可以使用如下命令：

```
dsquery user "OU=Marketing,DC=ny,DC=cpandl,DC=com" | dsmod group
"CN=AllSales,OU=Sales,DC=ny,DC=cpandl,DC=com" -rmmbr

dsquery user "OU=CustSupport,DC=ny,DC=cpandl,DC=com" | dsmod group
"CN=AllSales,OU=Sales,DC=ny,DC=cpandl,DC=com" -rmmbr
```

其中，第一条命令获取Marketing OU中所有用户列表，之后将其作为输入信息传递给DSMOD GROUP，用来从AllSales组中移除这些用户。第二条命令获取CustSupport OU中所有用户列表，之后将其作为输入信息传递给DSMOD GROUP，用来从AllSales组中移除这些用户。

提示 如果两个用户列表不能与AllSales组中的成员准确匹配，就会带来问题。比如，如果新的市场用户开始工作，并且已经被添加到Marketing OU，但尚未赋予对Sales信息的访问权限，也没有在AllSales组中。这种情况下，DSMOD GROUP发现第一个不匹配时，就会退出并报告错误信息。但如果是一个很微小的错误，你可能并不希望退出该命令。为此，可以使用-C参数，这一参数会在报告错误的同时保证该命令继续执行。

与使用-Addmbr参数类似，你也可以直接指定待删除对象DN。比如，假定需要从AllSales组中移除SalesLocal组与SalesGlobal组，则可以使用如下命令：

```
dsquery group -name AllSales | dsmod group
-rmmbr "CN=SalesLocal,OU=Sales,DC=ny,DC=cpandl,DC=com"
"CN=SalesGlobal,OU=Sales,DC=ny,DC=cpandl,DC=com"
```

注解 上面的命令中，由于页面排版的原因，可能看不清楚，但在每个组的DN之间是有空格的。空格是必要的，分隔之后，每个组的DN才能分别正确解释。

3. 替换组中所有成员

与GUI工具相比，命令行中可以一次对组中所有成员进行替换。比如，如果AllUsers组中的组成员不是最新更新的，并且手工添加或移除组成员比较困难，则可以对其进行一次性替换。

你可以使用DSMOD GROUP与-Chmbr参数来替换现存的组成员，该参数可以接受来自DSQUERY USER的输入，或者空格分隔的DN列表。比如，如果需要替换现有的组成员列表，并将组织中的所有用户添加到AllUsers组，则可以使用如下命令：

```
dsquery user -name * | dsmod group
"CN=AllUsers,CN=Users,DC=seattle,DC=cpandl,DC=com" -chmbr
```

上面的命令中，DSMOD GROUP首先移除所有成员，之后添加传递过来的对象。如果命令中的任一部分出错，则命令会失败，也不会产生任何实际的改变。

注解 尽管通过-C参数，可以确保即便发生错误，操作仍能继续进行，但这会导致组内存在空成员关系。由此可能导致的问题是，DSMOD GROUP可以没有任何问题地移除当前组成员，但无法添加成员。成员的移除只需要具备适当的管理权限，但成员的添加则依赖于所提供的输入信息。

15.6.5 移动组账号

与用户账号类似，组账号也可以移动到我当前所在域内的不同容器或OU中。为此，你需要使用DSMOVE命令指定组账号的当前DN，并使用-Newparent参数指定组账号的新位置或父DN。比如，如果需要将ProdDev组从Users容器中移动到Developers组织单元，你需要指定组账号的DN，比如“CN=ProdDev,CN=Users,DC=cpandl,DC=com”，并提供新位置的父DN，比如“OU=Developers,DC=cpandl,DC=com”。相关的命令类似于如下的格式：

```
dsmove "CN=ProdDev,CN=Users,DC=cpandl,DC=com"
-newparent "OU=Developers,DC=cpandl,DC=com"
```

通过使用DSQUERY GROUP命令，可以将获取的组DN传递给DSMOVE作为输入信息，可以减少一些键入工作，如下面实例所示：

```
dsquery group -name "ProdDev" | dsmove
-newparent "OU=Developers,DC=cpandl,DC=com"
```

上面的命令中，组账号DN，“CN=ProdDev,CN=Users,DC=cpandl,DC=com”是从DSQUERY GROUP命令获取的，并用作DSMOVE的输入。

15.6.6 组账号重命名

与用户账号类似，组账号也包含安全标识符。因而，在改变组名之后，并不需要改变其对文件与文件夹等资源的访问权限。对组进行重命名时，改变的是其通常名。

你可以使用DSMOVE命令对组进行重命名，指定组DN，并使用-Newname参数指定新的通常名。比如，可以将组对象从ProDevs重命名为TechDevs，使用如下命令：

```
dsmove "CN=ProDevs,OU=Developers,DC=cpandl,DC=com" -newname "TechDevs"
```

你也可以使用DSQUERY GROUP命令获取组DN，参考如下实例：

```
dsquery group -name ProDevs | dsmove -newname "TechDevs"
```

这一命令中，使用DSQUERY GROUP命令获取ProDevs组DN，之后使用DSMOVE命令对其进行重命名。

由于对组进行重命名并不会改变Windows 2000之前的组名或与组相关联的描述信息，因此，你还需要改变这些属性。为此，可以使用DSMOD GROUP命令，并使用-Samid参数设置Windows 2000之前的组名，使用-Desc参数为其设置描述信息，参考如下实例：

```
dsquery group -name TechDevs | dsmod -samid techdevs  
-desc "Technical Developers Group"
```

上面命令中，将Windows 2000之前的组名改变为techdevs，描述信息改变为Technical Developers Group。

15.6.7 删除组账号

要从活动目录中永久性删除组账号，可以使用DSRM命令。大多数情况下，你需要删除的是某一个指定的组，而不是很多组，比如所有组名以M引导的组。如果是这种情况，你需要将待删除组的DN传递给DSRM命令，比如：

```
dsrm "CN=AllSales,OU=Sales,DC=chicago,DC=cpandl,DC=com"
```

默认情况下，DSRM会弹出提示信息询问是否确认删除。如果不希望看到提示信息，可以使用-Noprompt参数，比如：

```
dsrm "CN=AllSales,OU=Sales,DC=chicago,DC=cpandl,DC=com" -noprompt
```

有些情况下，你可能需要一次移除几个组。比如，由于公司范围的重组，你可能发现市场部职能已经外包了，不再需要市场相关的组。如果这些相关组的组名都是以关键词Marketing引导，你可以通过如下命令删除这些组：

```
dsquery group -name Marketing* | dsrm -c
```

上面的命令中，将所有组名以Marketing引导的组的DN传递给DSRM命令，-C参数的作用是在发生错误时保证操作继续进行。

警告 即便输入信息是从DSQUERY GROUP命令传递来的，你也不能由DSRM命令自己使用。比如，你不能使用dsquery group -name Marketing* | dsrm这样的命令，原因是命令行仍然期望对象的DN或其他参数跟随在DSRM命令之后。因此，你还需要使用一些相关参数，使用-C是最安全的，因为这会保证在有错误发生时DSRM仍可以继续执行。而如果使用-Noprompt参数，则DSRM会在不提示用户的情况下进行删除操作，这可能会导致删除一些本不应该删除的组，但没有办法恢复。

Part 5

第五部分

使用命令行管理网络

本部分内容

- 第 16 章 管理网络打印机与打印服务
- 第 17 章 TCP/IP 网络的配置、管理与故障排除

大多数组织所拥有的打印机都是多样化的，包括一些高容量、高成本的打印机，也包括一些低容量、低成本的打印机。典型情况下，高容量打印机用于处理多个用户繁重的、日常的打印任务，低容量、低成本打印机用于处理小型组或单独用户的打印任务。不管哪种用途，打印服务器都需要足够的内存与处理能力来提供打印服务。在高容量的打印环境，或者经常性地打印很多非常大、非常复杂文档的环境中，需要对打印服务器进行特殊配置，或者使其专门提供打印服务。其他对打印只有一般性需求的环境中，打印服务器并不会非常昂贵，也不会是专门的计算机。实际上，很多打印服务器都是标准的桌面系统，同时还提供其他网络服务。要记住的是，Windows Server 2008与Windows Vista为文件共享赋予了比打印共享更高的优先级。因此，如果某系统同时处理这两种共享服务，则打印服务将会被抑制，以便先提供文件服务，从而防止出现文件访问性能问题。

打印服务器必须拥有足够的磁盘空间，用来处理打印任务。所需要的磁盘空间总量取决于打印任务的大小与打印队列的长度。为获得最佳性能，打印机缓冲池文件夹应该存储在专用的磁盘驱动器上（不用于打印缓冲之外的任何其他用途）。打印服务管理关键的一个部分是维护。为正确维护与支持打印服务，应该注意及时追踪打印缓冲池信息与使用情况的统计资料，这些信息有助于确定打印服务的执行情况。尽管你主要关注的可能是打印相关的性能问题，但你会发现，有几个命令行工具对打印服务器维护以及打印机故障排除也是有用的，本章将对这些工具进行讲解与讨论。

16.1 获取打印机的支持信息与故障排除信息

打印机在采购与部署时通常并未过多考虑如何使用。比如，公司内的某个职员觉得某个办公室需要一个打印机，就会采购并安装一个打印机。有时候这项工作并不是由管理员完成的，所以在管理员管理打印时，就会比较盲目。但不管打印机是如何获取的，管理员或支持小组都应该维护每一台打印机的配置信息，包括有哪些可用的驱动程序、使用的是哪种驱动程序等。作为管理员，应该定期检查打印机的任务繁忙程度，以及是否当前正在处理打印任务。也可能需要追踪打印机状态、打印队列中任务数以及其他有助于确定存在问题的重要信息。很多情况下，这些信息对于容量规划也是有用的。

16.1.1 在命令行中操作打印机

与操作其他组件或硬件相比，在命令行中操作打印机存在一些不同，主要原因在于使用的工具存储在子目录中，并且这些子目录不是默认情况下命令路径的组成部分。为此，你或者需要切换到这些

子目录中使用相关工具，或者对命令路径进行更新——2.2.1节对这一问题进行了讨论。在%SystemRoot%\System32\Printing_Admin_Scripts的场所特定的文件夹中，比如C:\Windows\System32\Printing_Admin_Scripts\en-us，你会发现其中存在如下一些Windows脚本。

- ❑ prncnfg.vbs。用于列出与管理打印机配置设置信息。
- ❑ prndrvr.vbs。用于列出、安装与管理打印机驱动程序。
- ❑ prnjobs.vbs。用于列出与管理打印队列中的打印任务。
- ❑ prnmngr.vbs。用于安装、列出与移除打印机。
- ❑ prnport.vbs。用于添加、配置与移除打印机使用的TCP/IP端口。
- ❑ prnqctl.vbs。用于管理打印队列。
- ❑ pubprn.vbs。用于在活动目录中发布打印机。

如果是第一次在命令行中操作脚本，或者已将WScript设置为主脚本宿主，则需要将CScript设置为默认的脚本宿主。为此，需要在命令行中键入**cscript //h: cscript //s**。设置之后，就可以操作命令行脚本宿主，而非图形界面的脚本宿主。要记住的是，脚本宿主是以每个用户为单位进行设置的。因而，如果需要以某个用户身份运行一个脚本，但该用户可能尚未将CScript设置为默认的脚本宿主，为避免这一情况，可以在脚本中添加一行：**cscript //h: cscript //s**。

对上面的每一个脚本，通过-S参数，可以指定要操作的远程计算机；通过使用-U与-W参数，可以指定登录凭据。其中，-U用于指定远程登录的用户账号，-W用于指定该账号的口令。下面的实例中，将远程计算机设置为PrintServer43，并使用WilliamS的登录凭据：

```
-s PrintServer43 -u WilliamS -w Rover
```

注解 与大多数命令类似，你也可以同时指定用户域与用户账号，其格式为Domain/User。

另一个有用的工具是打印机备份与迁移工具（Printbrm.exe）。在为远程服务器管理安装Print Services工具时，或者为Windows服务器添加Print Services角色时，会在%SystemRoot%\System32\Spool\Tools文件夹中发现Printbrm，使用Printbrm可以完成如下一些任务。

- ❑ 列出打印机的配置信息摘要。
- ❑ 备份与恢复打印服务器的配置。
- ❑ 恢复时将LPR端口转换为TCP/IP端口。
- ❑ 将打印机与打印队列从某计算机迁移到其他计算机。
- ❑ 在活动目录中发布所有可用的打印机。

由于Printbrm是一个命令行工具，因此不需要通过脚本宿主执行。使用Printbrm时，可以使用-S参数指定待操作的远程主机。然而，由于无法指定可替代的登录凭据，你应该使用增强的命令提示符，并以一个具有操作打印机、打印驱动程序、打印队列权限的用户身份登录。

16.1.2 追踪打印驱动程序与打印机信息

要更好地了解打印服务器上打印机的配置与使用方式，你可能需要追踪其上安装的打印机的详细信息。Prndrvr就是一款可用于获取已安装打印机及其驱动程序信息的工具。使用Prndrvr与-L参数，可以列出本地计算机上安装的所有打印机及其各自的打印驱动程序配置。如命令清单16-1中所示，所列出的打印驱动程序与打印机信息是非常详细的。

命令清单16-1 Pnndrvr -l的输出信息

```

Server name PrintServer43
Driver name magicolor 2300 DL,3,Windows NT x86
Version 3
Environment Windows NT x86
Monitor name MLMON__B.DLL
Driver path C:\Windows\system32\spool\DRIVERS\W32X86\3\MIMFN5_B.DLL
Data file C:\Windows\system32\spool\DRIVERS\W32X86\3\MSDMLT_B.SDD
Config file C:\Windows\system32\spool\DRIVERS\W32X86\3\MNT5UI_B.DLL
Help file C:\Windows\system32\spool\DRIVERS\W32X86\3\MSDMLT_B.HLP
Dependent files
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSPL32_B.EXE
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSP00L_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MIMFPR_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MIMF32_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSDIMF_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MQDPRT_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSD32__B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSR32__B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MDDM32_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MCMM__B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MICM__B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MGDI32_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MDDMUI_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MTAG32_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MLTSRV_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSUMLT_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSUMLT_B.INI
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSDMLT_B.DLL
C:\Windows\system32\spool\DRIVERS\W32X86\3\MICM6__B.ICM
C:\Windows\system32\spool\DRIVERS\W32X86\3\MICM12_B.ICM
C:\Windows\system32\spool\DRIVERS\W32X86\3\MICM24_B.ICM
C:\Windows\system32\spool\DRIVERS\W32X86\3\MICM6L_B.ICM
C:\Windows\system32\spool\DRIVERS\W32X86\3\MICM12LB.ICM
C:\Windows\system32\spool\DRIVERS\W32X86\3\MICM24LB.ICM
C:\Windows\system32\spool\DRIVERS\W32X86\3\MSEP01_B.SEP
C:\Windows\system32\spool\DRIVERS\W32X86\3\MUINST_B.EXE
C:\Windows\system32\spool\DRIVERS\W32X86\3\MUNZ__B.UNM

```

检查这些信息可以发现，其中包含了如下的一些信息。

- 打印机驱动程序名，比如magicolor 2300 dl。Windows使用打印机驱动程序名来追踪打印机驱动程序，打印机使用的驱动程序应该与打印机的实际类型相匹配。在上面的实例中，打印机属于Minolta Magicolor 2300 DL系列打印机。打印文档时，用于打印的应用程序会使用打印机驱动程序将文档转换为物理打印设备可以识别的文件格式。如果打印机出现问题，并怀疑是因为加载了错误的驱动程序，这将是最好的标志。
- 打印机驱动程序模式。打印机驱动程序或者运作在类型2（内核）模式，或者运作在类型3（用户）模式。在相应的输出信息中，内核模式标记为Version 2，用户模式标记为Version 3。在内核模式下。打印机驱动程序的运作方式与其他由操作系统运行的程序类似；在用户模式下，打印机驱动程序的运作方式与其他由用户运行的程序类似。典型情况下，内核模式下的终止错误信息比用户模式下的终止错误信息更为详细。然而，如果打印机驱动程序本身存在问题，

则运行在内核模式下更容易造成系统的不稳定。在Windows Server 2008与Windows Vista下，更倾向于选择使用用户模式的打印机驱动程序，以便确保操作系统自身的稳定性。

- **可用的打印机环境。**共享打印机时，Windows会自动将驱动程序设置为可用，以便用户在第一次连接该打印机时可以下载。典型情况下，只有类型3 X86驱动程序是默认可用的。类型3 X86驱动程序可用于32位版本的Windows。为使得驱动程序适用于其他环境，需要安装并激活新的打印机驱动程序。比如，如果所在组织拥有X64或IA64计算机，就需要为这些计算机安装适当的打印机驱动程序。
- **使用的打印监视器。**每一个打印设备都有相关联的打印监视器。支持双向打印的打印机有一个语言监视器，用于处理打印机与打印缓冲池之间的双工通信，还有一个端口监视器，用于控制与打印机相连的I/O端口，所有这些统称为打印设备的打印监视器。如果打印机有相关联的语言监视器，则该监视器的名字被指定为与文件名相同，只是没有.dll扩展名。如果没有语言监视器，则名字值指定为Null，或者不指定。待打印文档到达打印机栈的栈顶后，打印监视器负责将其发送到相关的打印设备，即实际完成打印工作的物理打印机。大多数打印设备都有自己的打印监视器，是由打印设备的制造商创建的。Windows也有自己默认的打印监视器。要使得打印设备完成实际的打印任务，打印监视器是必需的。如果打印监视器损坏或者找不到，则需要重新安装。
- **打印缓冲池DLL与相关的数据文件。**打印缓冲池的特定DLL文件是由驱动程序路径指定的。缓冲池有相关联的数据、配置与帮助文件。打印缓冲池用于存储用户需要打印处理器处理的文档。打印处理器的作用是创建原始的、必要的打印数据，用来提供给打印设备。打印设备再将这些数据回传给打印缓冲池，以便其在缓冲池中排队等待打印。
- **打印机驱动程序栈文件。**与特定打印机驱动程序相关联的所有栈文件都是以独立文件的形式列出的。待打印文档的路由（使用打印路由器）是从打印缓冲池到打印机栈的，也称为打印队列。出现在打印队列中之后，待打印文档就称为打印任务，这意味着该文档已经成为打印缓冲池需要处理的任务。

如果需要使用Pnmdrvr返回远程打印服务器与网络打印机的驱动程序信息，可以使用-S参数，其后跟随服务器的域名，比如：

```
pnmrvr -s corpserver01
```

上面的命令中，检查的是CorpServer01的打印机驱动程序信息。

Pnmdrvr命令的输出中，缺少一些详细资料，无法完整反映打印机配置的相关信息。要获取更多打印机相关信息，可以使用Printbrm命令与-Q参数。尽管Printbrm -Q的输出与Pnmdrvr -l的输出类似，但可以提供一些重要的附加信息。参考如下实例输出信息：

```
Operation mode: query
Target server: local machine
Queue publish mode: none
Overwrite Mode: keep existing settings
```

```
LISTING PRINT QUEUES
hp laserjet 9500 on second floor
magicolor 2300 main floor
Adobe PDF
```

```
LISTING PRINTER DRIVERS
```



```
hp laserjet 9500 series, Windows NT x86, HP_PRNMON.DLL
magicolor 2300 DL, Windows NT x86, MLMON_B.DLL
Adobe PDF Converter, Windows NT x86, None
```

LISTING PRINT PROCESSORS

```
hpzpp1hn Windows NT x86 hpzpp1hn.dll
MIMFPR_B Windows NT x86 MIMFPR_B.DLL
```

LISTING PRINTER PORTS

```
192.168.0.90, TCP
192.168.1.90, TCP
```

Displaying print hierarchy.

```
hp laserjet 9500 on second floor
    hp laserjet 9500 (Windows NT x86) #1
        192.168.1.80 #1
magicolor 2300 main floor
    magicolor 2300 DL (Windows NT x86) #1
        192.168.1.90 #1
Adobe PDF
    Adobe PDF Converter (Windows NT x86) #1
Unassociated:
    192.168.0.70 #0
```

上面的输出信息中，可以发现如下一些信息。

- ❑ 所有打印队列的列表，分别列出了打印队列名。
- ❑ 已安装打印机的打印驱动程序列表，列出了打印驱动程序名、激活的驱动程序环境以及相关关联的打印监视器。
- ❑ 已配置的网络打印机端口列表，列出了相关的IP地址与类型。
- ❑ 计算机上打印体系列表，用于将打印队列与相关的打印驱动程序（也可能还有打印机端口）进行关联。

注解 打印机可以使用LPT、COM或USB端口直接连接到打印服务器，网络上的打印机通常还设置了IP地址与TCP端口。

16.1.3 获取用于容量规划与故障排除的打印详细统计资料

通过追踪打印队列信息与使用情况的统计资料，有助于回答如下一些重要问题。

- ❑ 打印服务器的平均繁忙程度？
- ❑ 打印任务的平均大小？
- ❑ 打印队列中排队等待的打印任务数？
- ❑ 当前打印机状态？
- ❑ 打印缓冲池运行了多久？
- ❑ 打印机运行了多久？
- ❑ 打印服务器运行了多久？

这些问题之所以重要，是因为通过对这些问题的解答，可以更积极地管理与维护组织内的打印服务，也可以对未来的打印需求进行规划。作为管理员，不应该仅仅满足于对这些问题进行解答，而是应该更进一步，以便为组织内的打印用户提供更好的打印服务。

用于追踪打印缓冲池信息与使用情况统计信息的关键工具是Print Queue这一计数器对象，该对象可以通过TYPEPERF命令进行访问。这一性能计数器对象包含了很多性能计数器，可用于追踪打印队列与使用情况统计资料。如第6章到第9章中所讨论的，在命令行中，可以使用很多技术来操作和自动监控性能对象。

操作Print Queue对象时，你可能需要追踪_Total计数器实例，用来确定打印服务器的整体繁忙程度，也可能需要追踪单独的计数器实例，用来确定某个特定打印队列的繁忙程度。可用于确定使用情况统计资料的重要计数器包括下面列举的。

- **Bytes Printed/sec**。列出每秒钟打印的字节数，用来确定打印机处理的数据量以及每台打印机的繁忙程度。通过将Bytes Printed/sec与打印服务器的开机时间相比较，可以了解打印机每小时或每天处理的数据量。

注解 有些打印机的配置中，打印任务是在排队后保存的，使得用户可以从打印队列（而不是从应用程序）将文档重提交给打印机。如果需要将打印机配置为保存打印任务，就应该监视Bytes Printed/sec与打印任务的总数。这些信息有助于确定需要多大的磁盘空间，以便维护打印服务，并了解需要从打印队列中清除旧打印任务的频率。

提示 大多数打印机拥有自己的内部内存。理想情况下，你可能希望该内存足够大，以便整个打印任务可由打印设备自身完成。如果打印机需要经常性地处理大的或复杂的打印任务，可以考虑为打印设备添加内存。你需要通过打印机的配置页面（可以在打印机自身进行打印）来确定其已安装的内存大小。

- **Jobs**。展示排队等待的打印任务数。典型情况下，繁忙的打印机都会有几个排队等待的打印任务，尤其在峰值打印时间段。然而，如果频繁地发现有很多打印任务需要排队等待，则可能说明该打印机处于任务过载状态。为解决这一问题，可以让用户知道还有哪些可用的打印机，或者为某些用户设置不同的默认打印机。
- **Jobs Spooling**。列出当前缓冲池中的打印任务数（这些任务稍后将进入打印队列），实际上也是入栈的任务数。
- **Max Jobs Spooling**。列出当前缓冲池中的峰值打印任务数。
- **References**。列出打印队列当前开放的句柄数，开放句柄可以来自当前没有处于活跃打印状态的用户，每个开放的句柄都会占用资源。
- **Max References**。列出打印队列的峰值开放句柄数。
- **Total Jobs Printed**。展示自打印服务器上次重启以来处理的打印任务数，该值可以相对表示出某打印机的繁忙程度。通过将打印任务数总量与打印服务器的开机时间相比较，可以准确确定打印机的繁忙程度。
- **Total Pages Printed**。展示自上次重启以来打印队列中打印的页面数，该值可以相对表示出某打印机的繁忙程度。

示例16-1展示了一个使用TYPEPERF命令获取企业内多台打印服务器相对打印负载快照的实例。该实例中，使用名为Perf.txt的计数器文件指定待追踪的计数器。除打印队列计数器之外，还对System

对象的System Up Time计数器进行了追踪,以便确定自计算机上次重启以后运行的时间(以秒为计数单位)。从每一台打印服务器上收集一个示例,将输出信息保存到一个名为SaveData.txt的文件中。如果将这些数据导入到电子表格中,或者将其转换到Word文档中的表格中,就可以更清晰地观察这些输出信息,并准确理解每台打印服务器的繁忙程度。

示例16-1 获取打印服务器的使用情况统计资料

Command line

```
typeperf -cf c:\printers\perf.txt -o c:\printers\savedata.txt -so 1 -y
```

Source for perf.txt

```
\\printserver14\system\System Up Time
\\printserver14\print queue(_Total)\Bytes Printed/Sec
\\printserver14\print queue(_Total)\Jobs Spooling
\\printserver14\print queue(_Total)\Max Jobs Spooling
\\printserver14\print queue(_Total)\Jobs
\\printserver14\print queue(_Total)\References
\\printserver14\print queue(_Total)\Max References
\\printserver14\print queue(_Total)\Total Jobs Printed
\\printserver14\print queue(_Total)\Total Pages Printed
\\printserver21\system\System Up Time
\\printserver21\print queue(_Total)\Bytes Printed/Sec
\\printserver21\print queue(_Total)\Jobs Spooling
\\printserver21\print queue(_Total)\Max Jobs Spooling
\\printserver21\print queue(_Total)\Jobs
\\printserver21\print queue(_Total)\References
\\printserver21\print queue(_Total)\Max References
\\printserver21\print queue(_Total)\Total Jobs Printed
\\printserver21\print queue(_Total)\Total Pages Printed
\\printserver32\system\System Up Time
\\printserver32\print queue(_Total)\Bytes Printed/Sec
\\printserver32\print queue(_Total)\Jobs Spooling
\\printserver32\print queue(_Total)\Max Jobs Spooling
\\printserver32\print queue(_Total)\Jobs
\\printserver32\print queue(_Total)\References
\\printserver32\print queue(_Total)\Max References
\\printserver32\print queue(_Total)\Total Jobs Printed
\\printserver32\print queue(_Total)\Total Pages Printed
```

Sample output

```
"(PDH-CSV 4.0)","\\printserver14\system\System Up Time","
\\printserver14\print queue(_Total)\Bytes Printed/
Sec","\\printserver14\print queue(_Total)\Jobs Spooling","
\\printserver14\print queue(_Total)\Max Jobs Spooling","\\printserver14\
print queue(_Total)\Jobs","\\printserver14\print queue(_Total)
\References","\\printserver14\print queue(_Total)\Max References","
\\printserver14\print queue(_Total)\Total Jobs Printed","
\\printserver14\print queue(_Total)\Total Pages Printed"
"10/12/2009 08:20.509","15535.955367","96.827000","3.000000","19.000000",
"8.000000","93.000000","151.000000","267.000000","2413.000000"

"(PDH-CSV 4.0)","\\printserver21\system\System Up Time","
\\printserver21\print queue(_Total)\Bytes Printed/Sec",
```

```

\\printserver21\print queue(_Total)\Jobs Spooling", "
\\printserver21\print queue(_Total)\Max Jobs Spooling", "
\\printserver21\print queue(_Total)\Jobs", "
\\printserver21\print queue(_Total)\References", "
\\printserver21\print queue(_Total)\Max References", "
\\printserver21\print queue(_Total)\Total Jobs Printed", "
\\printserver21\print queue(_Total)\Total Pages Printed"
"10/12/2009 08:21.002", "2487384 875323" "124 393923" "17 000000", "
39.000000", "12.000000", "165.000000", "223.000000", "17897.000000", "
35672.000000"

"(PDH-CSV 4.0)", "\\printserver34\system\System Up Time", "
\\printserver34\print queue(_Total)\Bytes Printed/Sec", "
\\printserver34\print queue(_Total)\Jobs Spooling", "
\\printserver34\print queue(_Total)\Max Jobs Spooling", "
\\printserver34\print queue(_Total)\Jobs", "
\\printserver34\print queue(_Total)\References", "
\\printserver34\print queue(_Total)\Max References", "
\\printserver34\print queue(_Total)\Total Jobs Printed", "
\\printserver34\print queue(_Total)\Total Pages Printed"
"10/12/2009 08:21.535", "96375.673823", "24.975632",
"2.000000", "7.000000", "3.000000", "42.000000", "67.000000"
"514.000000", "5785.000000"

```

上面的实例中，对3台打印服务器的打印队列进行了检查，并假定每台打印服务器有一个主打印队列、单台打印服务器上所有活跃打印队列针对的是同一台物理打印机。检查示例中PrintServer14的输出信息，可以确定如下一些事实。

- 该服务器平均每小时处理62个打印任务。因为打印池中共计267个打印任务，服务器开机时间为4.3个小时（259分钟），用Total Jobs Printed值除以System Up Time值（以小时为计数单位，而非分钟），就可以获得这个结果。
- 打印任务的平均长度为2页，这是通过将打印的页面总数除以打印任务总数得到的。
- 当前，该打印服务器缓冲池中有3个活跃的打印任务，打印队列中有8个打印任务，缓冲池中打印任务峰值为19。

检查示例中PrintServer21的输出信息，可以确定如下一些事实。

- 该服务器平均每小时处理26个打印任务。因为打印池中共计17,897个打印任务，服务器开机时间为691个小时，用Total Jobs Printed值除以System Up Time值（以小时为计数单位，而非分钟），就可以获得这个结果。
- 打印任务的平均长度为2页，这是通过将打印的页面总数除以打印任务总数得到的。
- 当前，该打印服务器缓冲池中有17个活跃的打印任务，打印队列中有12个打印任务，缓冲池中打印任务峰值为39。

检查示例中PrintServer34的输出信息，可以确定如下一些事实。

- 该服务器平均每小时处理19个打印任务。因为打印池中共计514个打印任务，服务器开机时间为27个小时，用Total Jobs Printed值除以System Up Time值（以小时为计数单位，而非分钟），就可以获得这个结果。
- 打印任务的平均长度为11.25页，这是通过将打印的页面总数除以打印任务总数得到的。
- 当前，该打印服务器缓冲池中有2个活跃的打印任务，打印队列中有3个打印任务，缓冲池中

打印任务峰值为7。

从上面信息可以看出，可以发现当前的打印环境处于繁忙状态，有较重的打印负载。如果在几个时间间隔内检查这些统计信息，并在几次重启printers/spooler之后都看到类似的统计信息，则需要对其进行检查或重新配置。因为这些信息表明打印机处于非常繁忙的状态，考虑到大多数打印机实际上只有在上班时间（12小时，甚而8小时）才会有人使用，这一情况就更需要关注。

对于这一水平的使用情况，你可能需要密切关注服务器的使用与性能信息，如第7章中所讨论的。你可能还需要深入挖掘使用情况的统计资料，查看每台打印机配置的详细情况。在对系统性能与使用情况进行了足够时间间隔的监测后，你会发现如下一些事实。

- 打印服务器需要添加更多的内存，因为在任意给定的时间点上都有大量的打印任务需要处理。
- 需要更强的处理能力，因为平均处理的打印任务数较大。
- 需要更大的磁盘空间或者用于缓冲池文件夹的专用磁盘驱动器。

Print Queue对象还包括几个附加的性能计数器，有助于常规监控，包括下面列举的。

- **Job Errors**。列出自上次重启以来打印队列中的任务错误数。在将打印任务传递给打印机时，如果中间出现问题，就会导致任务错误。如果任务错误数相对较多，则可能表明存在网络故障，或是网卡问题。
- **Not Ready Errors**。列出自上次重启以来打印队列中的打印机尚未就绪错误。如果打印机等待用户输入，或者没有做好打印的准备，就会出现这一错误。
- **Out Of Paper Errors**。列出自上次重启以来打印队列中的出纸错误。如果打印机频繁出现这一错误，则说明打印纸没有正确配套，或者需要一个附加的进纸盒。

上面的示例中，给出的打印系统是相对简单的，通过常规性地监控与适当调整，就可以确保打印服务的正确运行。如果需要自动监控，可以创建一个脚本，将使用情况统计资料写入到一个日志文件中，并将该脚本设置为计划任务，使其在适当的时间间隔运行，第9章中对这些技术进行了讨论。

要记住的是，通常需要的是以几天为周期对打印机使用情况进行监控，以便确定是否需要进行必要的升级或修改。进行升级或修改时，你需要终止打印缓冲池，完成之后再重启。

16.2 管理打印机

在命令行中，可以使用Prnmngr工具安装与管理打印机。通过Prnmngr，可以操作物理连接到计算机上的打印设备，但只有登录到该计算机的用户才可以进行操作，对这些用户而言，打印设备称之为本地打印设备，对于那些通过网络进行操作的打印设备，则称之为网络打印设备。本地打印机与网络打印机的关键区别在于，本地打印机不能共享。在网络上共享打印机时，需要使用某台计算机作为该打印机的宿主机，该计算机称之为打印服务器。

打印服务器的主要任务是将打印设备共享在网络上，并处理打印缓冲池。通过使用打印服务器，可以很便利地对打印队列进行集中式管理，而不需要在客户端系统上安装打印机驱动程序。然而，并不是一定要使用打印服务器。实际上，用户也可以直接连接到网络上的打印机，之后就像对连接到用户计算机上的本地打印机一样进行操作。这种方式下，每个用户有自己的打印队列，必须分别进行管理。

在计算机上安装打印机时，实际上是配置一个打印队列，以便对打印任务进行路由排队，并在合适的时间将其发送到物理打印设备。因此，讨论安装打印机与配置打印机，实际上是讨论安装与配置

打印队列，以便打印任务发送到物理打印设备。

如果需要安装或配置打印机，需要具备适当的管理员特权。在域内，这意味着必须是管理员组、Print Operators组或Server Operators组的成员。如果只是需要连接并使用打印机，则不需要具备管理员特权，只需要具备适当的访问许可权限就可以。

16.2.1 安装物理连接的打印设备

物理连接的打印设备直接连接在计算机上，可以配置为本地打印设备或网络打印设备。本地打印设备只对登录到该计算机的用户是可用的，网络打印设备则作为共享资源，网络上具备适当权限的用户都可以访问并使用。配置时，使用适当的串口、并口或USB接口将打印设备物理连接到服务器上。如果配置的是网络打印机，则此计算机充当打印服务器。对即插即用式打印机，登录计算机后，只需要将打印机插到计算机上，就可以自动地识别、安装与配置打印机。

你可以使用Prnmngr命令与如下一些参数，手动安装本地打印机。

- **-A AddPrinter**。指定要添加或安装本地打印机。
- **-P PrinterName**。为打印机指定一个打印机名。在控制面板的打印机页面或命令行中对打印机进行操作时，看到的就是这个打印机名。
- **-M PrinterModel**。指定打印机型号。型号必须是由制造商指定的准确的型号，型号决定了打印机应该使用的驱动程序。
- **-R PrinterPort**。指定打印机应该连接到的端口号。端口可以是并口，比如LPT1:、LPT2:、LPT3:。也可以是串口，比如COM1:、COM2:、COM3:。还可以是USB端口，比如USB001。

注解 设置打印机名与型号时，字母的大小写与在命令提示符与对话框中显示的是一样的。然而，尽管设定和显示时这些名称能用大小写，但实际上并不区分大小写。也就是说，在Windows看来，centralcolorlaser与CentralColorLaser是一样的。

要配置物理连接的打印机，并不一定需要本地登录计算机，也可以远程进行安装与配置。为此，可以使用-S参数指定远程计算机名（需要为其添加本地打印机）。如果有必要，还可以分别使用-U参数与-W参数指定连接到远程计算机时使用的用户名与口令。

注解 在本地命令提示符进行操作时，不管是远程登录还是物理连接，都不能指定用户名与口令。如果尝试这样做，会返回错误消息“用户凭据不能用于本地连接。”

要了解如何使用Prnmngr命令，参考如下一些实例。

使用USB001配置一台HP 5500 Series InkJet打印机：

```
prnmngr -a -p "OfficeJetPrinter" -m "hp officejet 5500 series"
-r USB001
```

使用LPT1配置一台HP 1100 DN Series InkJet打印机：

```
prnmngr -a -p "BusinessJetPrinter" -m "hp businessjet 1100 series DN"
-r LPT1:
```

在cdesign09上使用USB001配置一台Epson Stylus Photo打印机:

```
prnmngr -a -p "PhotoPrinter" -m "epson stylus photo 1270 esc/p 2"
-r USB001 -s cdesign09
```

在mteam06上使用LPT1配置一台Epson Stylus Color打印机:

```
prnmngr -a -p "ColorPrinter" -m "epson stylus color esc/p 2"
-r LPT1: -s mteam06 -u wrstanek -w goldfish
```

如果打印机成功安装, Prnmngr会报告“已添加打印机”, 否则会报告“不能添加打印机”, 并描述发生的错误。最常见的错误是型号输入错误或设备型号未知, 这会导致Prnmngr报告打印机驱动程序未知。因此, 要确保输入了正确的打印机型号名。

注解 如果是计算机上安装的第一台打印机, 则会设置为默认的打印机, 但不会自动设置为共享的打印机。如果需要将其进行共享, 以便其他用户也可以使用, 可以参考16.4.2节所做的讲述。

提示 你可以为相同的打印设备创建附加的打印机, 唯一的要求是打印机名与共享名是唯一的。通过这种做法, 可以设置不同的属性, 用来满足不同的需求。比如, 一种配置为低优先级打印任务, 另一种配置为高优先级打印任务。

16.2.2 安装网络连接的打印设备

网络连接的打印设备直接通过网卡连接到网络上, 并配置为网络打印设备, 以便网络用户可以将它们作为共享打印设备对其进行访问。为此, 需要将打印机连接到网络上, 并为其配置适当的IP地址(或者从DHCP服务器获取IP地址), 具体操作可以参考制造商的打印机手册。

在打印机上配置TCP/IP后, 还需要在充当打印服务器的计算机上创建一个TCP/IP端口。以便网络用户通过该端口连接到打印机。之后, 就可以像安装物理连接的打印机一样对其进行安装, 唯一的区别是使用-R参数指定创建的TCP/IP端口, 而不是指定LPT、COM、USB等端口。比如, 如果创建了一个名为IP_192.168.10.15的TCP/IP端口, 则可以使用如下命令添加一个使用该端口的打印机:

```
prnmngr -a -p "CentralColorLaser" -m "magicolor 2300 d1" -r
IP_192.168.10.15 -s corpsvr03
```

上面的命令安装了一台Minolta QMS Magicolor彩色激光打印机, 该打印机使用TCP/IP端口。由于打印机是在CorpSvr03上配置的, 因此该计算机将充当此打印机的打印服务器, 但该打印机尚未设置为任意用户的默认打印机, 也没有设置为共享资源。如果需要将该打印机设置为共享, 以便其他用户可以使用, 可以参考16.4.2节的讲述。

16.2.3 列出计算机上配置的打印机

通过prnmngr -l命令, 可以列出本地计算机上配置的所有打印机。如果需要查看远程计算机上的这些信息, 可以使用-S参数, 其后跟随计算机名, 比如prnmngr -l -s corpsvr03。如果必要, 也可以分别使用-U参数与-W参数指定登录账号使用的用户名与口令。

上面命令的输出信息展示了打印服务器名(如果在本地计算机上使用, 则为空)以及其他一些关

于每台打印机配置的重要信息，参考下面的实例：

```
Server name corpsvr03
Printer name magicolor 2300 main for 5th floor
Share name magicolor
Driver name magicolor 2300 DL
Port name 192.168.1.92
Comment Main printer for the fifth floor.
Location 5/ne
Print processor MIMFPR_B
Data type IMF
Parameters
Attributes 2629
Priority 1
Default priority 0
Status Idle
Average pages per minute 8
Printer status Idle
Extended printer status Unknown
Detected error state Unknown
Extended detected error state Unknown
Number of printers enumerated 1
```

上面的实例中，打印机名、驱动程序名与端口名是在安装打印机时设置的，该打印机已设置为共享，以便域内其他用户可以用其完成打印任务。如果需要将该打印机移动到新的打印服务器上，真正需要注意的唯一信息是驱动程序名，大多数情况下该名与打印机型号是一致的。

16.2.4 查看与设置默认打印机

在命令提示符中键入 **prnmngr -g** 命令，可以显示当前登录用户的默认打印机。如果希望该用户使用其他默认打印机，可以键入 **prnmngr -t -p** 命令，其后跟随待设置的默认打印机名，比如：

```
prnmngr -t -p "magicolor 2300 DL"
```

如果成功执行，则 **Prnmngr** 会报告称打印机已经被设置为默认打印机。否则，**Prnmngr** 会报告错误信息。典型情况下，“Not Found” 错误说明输入了无效的打印机名。

16.2.5 打印机重命名

对打印机重命名是无法通过 **Prnmngr** 完成的，而是需要使用命令行工具 **Prncnfg**。对打印机重命名的语法格式为：

```
prncnfg -x -p CurrentPrinterName -z NewPrinterName
```

其中，**Prncnfg** 通过 **-X** 参数表明待执行的任务是对打印机重命名，之后使用 **-P** 参数指定当前打印机名，**-Z** 参数设定新打印机名，比如：

```
prncnfg -x -p "CentralColorLaser" -z "EngineeringPrinter"
```

如果该打印机存在，则 **Prncnfg** 会报告已对其进行了重命名，并设置了新打印机名。你也可以对远程计算机上的打印机重命名。为此，可以使用 **-S** 参数指定远程计算机名，比如：

```
prncnfg -x -s corpsvr03 -p "CentralColorLaser" -z "EngineeringPrinter"
```

上面的命令中，对CorpSvr03上的打印机重命名。然而，这一命令不允许设置登录账号。

16.2.6 删除打印机

Prnmngr提供了用于删除特定计算机上打印机的两种方法，你可以如下命令删除单独的打印机：

```
prnmngr -d -p PrinterName
```

比如：

```
prnmngr -d -p "magicolor 2300 DL"
```

如果输入了无效的打印机名，Prnmngr会报告由于没有发现打印机而无法删除。如果没有权限删除打印机，Prnmngr会报告用户登录凭据无法枚举打印机，你需要使用具备管理员特权的账号登录。注意的是，操作远程计算机时情况不是这样。操作远程计算机时，可以使用-U参数与-W参数指定登录账号及其口令，比如：

```
prnmngr -d -p "magicolor 2300 DL" -s corpsvr03 -u wrstanek  
-p goldfish
```

通过如下命令，可以删除计算机上所有打印机：

```
prnmngr -x
```

Prnmngr不会弹出提示信息询问是否确认操作，但会报告每个已删除的打印机，比如：

```
Deleted printer OfficeJet  
Deleted printer CentralPrinter
```

```
Number of local printers and connections enumerated 2  
Number of local printers and connections deleted 2
```

16.3 管理网络连接打印机的 TCP/IP 端口

要连接到网络连接的打印机，需要使用TCP/IP端口，这是使用Prnport创建与管理的。与Prnmngr类似，Prnport也是一个必须使用命令行脚本宿主运行的Windows脚本。

16.3.1 为打印机创建与改变 TCP/IP 端口

通过-A参数，可以通知Prnport要执行的任务是添加TCP/IP端口，之后使用-R参数指定端口名，使用-H参数指定打印机的IP地址。通常的做法是，端口名以连接的打印机IP地址为基础设置。比如，如果需要为IP地址为192.168.10.15的打印机配置端口，则可以将端口名设置为IP_192.168.10.15。

除端口外，还必须指定端口使用的输出协议。输出协议是使用-O参数设置的，或者是raw，或者是lpr。大多数打印机使用Raw协议。使用这种输出协议时，数据不经修改地使用指定的端口号发送到打印机。大多数情况下，这一端口号为9100，这也是之所以使用这一数值作为默认值的原因。当然，你也可以使用-N参数将端口号指定为不同的数值。使用LPR协议时，端口是与LPD（行式打印机守护进程）联合使用的，可以使用-Q参数设置打印队列名。

与大多数打印机配置命令类似，并不一定需要本地登录计算机来配置端口。如果需要对远程计算

机上的打印机端口进行配置，可以使用-S参数指定待操作的远程计算机名。如果必要，可以使用-U参数与-W参数指定连接到远程计算机的用户名与口令。如果登录域与当前域不同，用户名可以指定为Domain\Username的形式。参考如下一些实例。

添加一个端口，使用TCP Raw协议，通过9100端口连接到192.168.10.15:

```
prnport -a -r IP_192.168.10.15 -h 192.168.10.15 -o raw
```

添加一个端口，使用Raw协议，通过9500端口连接到10.10.1.50:

```
prnport -a -r IP_192.168.10.15 -h 10.10.1.50 -o raw -n 9500
```

添加一个端口，使用LPR输出，连接到172.20.18.2，将队列名设置为LPRQUEUE:

```
prnport -a -r IP_192.168.10.15 -h 172.20.18.2 -o lpr -q lprqueue
```

在CORPSVR03上添加一个端口，使用TCP Raw协议，通过9100端口连接到192.168.10.15:

```
prnport -a -r IP_192.168.10.15 -h 192.168.10.15 -o raw -s corpsvr03
```

如果成功创建，Prnport会报告“创建/更新端口”，否则会报告“无法创建/更新端口”，并描述发生的错误。

大多数网络连接的打印机也支持简单网络管理协议(SNMP)。为使打印机支持这一协议，必须使用-Me参数激活SNMP，并使用-Y参数设置SNMP community名，使用-l参数设置SNMP设备索引。典型情况下，community名设置为Public，这表明网络上的任意用户都可以使用与管理该打印设备。设备索引用于指定SNMP community中一个特定的设备。第一个设备索引为1，第二个设备索引为2，依此类推。

参考如下实例:

```
prnport -a -r IP_192.168.10.15 -h 192.168.10.15 -o raw -me -y public -i 1
```

注解 你可以使用-Md参数禁用SNMP。

上面的实例中，配置了一个TCP/IP端口，所用协议为TCP Raw，并通过9100端口连接到192.168.10.15。此外，还激活了SNMP，并将SNMP community名配置为public，设备索引则为1。

如果以后需要改变TCP/IP端口的配置，可以使用Prnport命令与-T参数。下面的实例中，使用-R参数指定待操作的端口，其他参数用于设置相关的属性值:

```
prnport -a -r MainPrinter -h 10.10.12.50 -o raw -md
```

这一命令中，试图修改MainPrinter这一TCP/IP端口，IP地址设置为10.10.12.50，输出协议设置为Raw，并禁用SNMP。

16.3.2 列出打印机使用的 TCP/IP 端口相关的信息

使用prnport -l命令，可以列出本地计算机上已配置的所有打印机TCP/IP端口。如果需要查看远程计算机的这一信息，可以使用-S参数，其后跟随计算机名，比如prnport -l -s corpsvr03。如果必要，也可以使用-U参数与-W参数指定登录账号的用户名与口令。

输出信息中会展示打印服务器名(如果是本地计算机则为空)，以及每个已配置端口的其他一些

重要信息。下面给出的是一个RAW端口相关信息的实例：

```
Server name
Port name IP_192.168.1.101
Host address 192.168.1.101
Protocol RAW
Port number 9100
SNMP Enabled
Community public
Device index 1
```

下面给出的是一个LPR端口相关信息的实例：

```
Server name
Port name IP_192.168.1.101
Host address 192.168.1.101
Protocol LPR
Queue crownnet
Byte Count Enabled
SNMP Enabled
Community public
Device index 1
```

注解 LPR端口信息会错误地展示字节计数是激活的。在激活的情况下，计算机会在将文档发送到打印机之前统计其中包含的字节数。大多数打印机不需要进行字节计数，字节计数会降低性能，因为打印时对文档中每一字节进行计数是一个非常耗时的过程。

16.3.3 删除打印机使用的 TCP/IP 端口

通过如下语法格式，可以删除打印机使用的各个端口：

```
prnport -d -r PortName
```

比如：

```
prnport -d -r IP_192.168.1.101
```

如果输入了无效的打印机名，Prnport会报告由于没有发现打印机而无法删除。如果没有许可权限删除打印机，Prnport会报告用户登录凭据无法枚举打印机，你需要使用具备管理员特权的账号登录。注意的是，操作远程计算机时情况不是这样。操作远程计算机时，可以使用-U参数与-W参数指定登录账号及其口令，比如：

```
prnport -d -r IP_192.168.1.101 -s corpsvr03 -u wrstanek -p goldfish
```

16.4 配置打印机属性

使用Prncnfg脚本与-T参数，可以查看并配置打印机属性。不管当前操作的是什么属性，Prncnfg都期望使用-P参数指定打印机。与大多数打印机配置命令类似，并不一定需要本地登录计算机来配置打印机属性。如果需要修改远程计算机上的打印机属性（而非打印机名），可以使用-S参数指定待操

作的远程计算机名。如果必要,可以使用-U参数与-W参数指定连接到远程计算机的用户名与口令。如果登录域与当前域不同,用户名可以指定为Domain\Username的形式。

16.4.1 添加注释与位置信息

通过为打印机添加注释与位置信息,可以使用户更便利地确定自己应该使用哪一台打印机。注释信息对打印机进行了常规意义上的描述,比如打印设备的类型与责任人。位置信息则描述了打印设备的物理位置。添加之后,这些信息会在打印机属性对话框中的“常规”选项卡中显示,也会在“打印”对话框(大多数应用程序中,选择了打印命令后会显示这一对话框)中显示。

为打印机添加注释与位置信息的语法格式如下:

```
prncnfg -t -p PrinterName -m "Comment" -l "Location"
```

上面的命令中,Prncnfg与-T参数的作用是声明要修改打印机属性,之后使用-M参数指定注释文本,使用-L参数指定打印机的位置信息,比如:

```
prncnfg -t -s corpsrv03 -p "CentralColorLaser" -m "Main Engineering  
Printer" -l "5th Floor SE"
```

如果成功执行,Prncnfg会报告已经对打印机进行了配置。如果不能执行,则可能是因为没有使用双引号对信息进行封装,也可能是因为忘记输入了其中的一个或几个参数。当然,并不要求一定同时输入注释与位置信息,你也可以分别设置这些信息。

16.4.2 共享打印机

在命令行中添加打印机之后,打印机并不会自动地设置为其他用户共享。如果需要共享打印机,必须专门使用Prncnfg命令对其进行配置。首先使用-T参数表明需要设置或改变打印机属性,使用-P参数指定待操作的打印机,之后使用-H参数设置共享名,并使用+Shared参数激活共享。为与Windows 2000之前的操作系统兼容,共享名在长度上应该为8个字符,并且不能包含空格。

由于位置信息可能并不总是可见的,因此,共享名中也可以包含用于指明打印机所在位置的相关信息,以便为用户节省一些时间。比如,如果某个打印机放置在第五层的东南角,则可以将其命名为FifthSE。参考如下实例:

```
prncnfg -t -s corpsrv03 -p "CentralColorLaser" -h "FifthSE" +shared
```

上面的命令中,将CorpSrv03的打印机CentralColorLaser设为共享,共享名为FifthSE。

如果需要移除打印机共享,可以使用-Shared参数。下面的实例中,移除了上面实例中配置的打印机共享:

```
prncnfg -t -s corpsrv03 -p "CentralColorLaser" -shared
```

16.4.3 在活动目录中发布打印机

通过将打印机相关信息发布到活动目录中,可以使得用户更容易找到可用的打印机。打印机在活动目录中发布之后,用户可以根据打印机的位置与功能进行搜索。比如,打印机是否在楼内第五层?是否可以彩打?

要对打印机发布进行配置,可以使用Prncnfg命令。如果需要将打印机发布到活动目录中,可以使

用-T参数表示要对打印机属性进行设置或修改，-P参数指定待操作的打印机，之后使用+Published参数表示将要发布打印机，或者使用-Published参数表示将从活动目录中移除打印机。

参考如下实例。

将CorpSrv03上的打印机CentralColorLaser发布到活动目录：

```
Prncnfg -t -s corpsrv03 -p "CentralColorLaser" +published
```

从活动目录中删除名为OfficeJet的本地打印机：

```
Prncnfg -t -p "OfficeJet" -published
```

无论是发布打印机还是移除打印机，Prncnfg都应该报告对打印机进行了配置。但如果打印机已经发布或移除，也不会报告错误。

16.4.4 设置分隔页并改变打印设备模式

分隔页可以用在打印任务的起始处，以便在繁忙的打印设备上找到某个文档。可以用于改变打印设备模式，比如打印设备是使用PostScript或打印机控制语言（PCL）。

分隔页存储在文件夹%SystemRoot%\System32中，Windows系统中定义了下面4种默认的分隔页。

- pcl.sep。将打印设备切换到PCL模式，并为每个文档打印一个分隔页。
- pscript.sep。将打印设备切换到PostScript模式，但并不打印分隔页。
- sysprint.sep。将打印设备切换到PostScript模式，并为每个文档打印一个分隔页。
- sysprintj.sep。将打印设备切换到PostScript模式，并为每个文档打印一个分隔页。实际上是Sysprint.sep的一个替代版，只是使用了不同版本的旗标文本。

通过Prncnfg，可以指定打印机应该使用上面的某个分隔页，或者文件夹%SystemRoot%\System32中的其他分隔页。使用-T参数表明正在设置或改变打印机属性，-P参数指定待操作的打印机，之后使用-F参数指定要使用的分隔页。

参考如下实例：

```
Prncnfg -t -s corpsrv03 -p "CentralColorLaser" -f sysprint.sep
```

上面的命令中，将CorpSrv03上的打印机CentralColorLaser配置为使用sysprint.sep。

如果需要终止使用分隔页，可以使用-F参数与" "，比如：

```
Prncnfg -t -s corpsrv03 -p "CentralColorLaser" -f " "
```

16.4.5 打印任务的调度与优先级设置

在命令行中，可以使用Prncnfg命令管理打印任务的优先级并对其进行调度。打印任务总是按照优先级顺序进行打印，1代表最低优先级，99代表最高优先级。高优先级的打印任务比低优先级的打印任务具有更高的优先打印权。如果某物理打印设备包含几个打印机（打印队列），则每一个高优先级的打印任务都可以优先得到打印。使用-T参数表明正在设置或改变打印机属性，-P参数指定待操作的打印机，之后使用-O参数设置打印任务的优先级：

```
prncnfg -t -p "EngineeringPrinter" -o 50
```

上面的命令中，将通过EngineeringPrinter传递给相关打印机的打印任务的优先级设定为50。比如，

如果某个Marketing打印机也配置为使用同一个物理打印设备，但优先级低于50，则Engineering打印任务总是会优先得以打印。

打印机或者总是可用的，或者只有在指定的时间段内可用。要设置打印机的可用性，可以使用-St参数与-Ut参数分别指定打印机可用与不可用的时间点，其中时间是以24小时时钟设置的，比如：

```
prncnfg -t -p "EngineeringPrinter" -st 0530 -ut 1930
```

上面的命令中，指定打印机在每天的上午5点30到下午的7点30是可用的。

16.4.6 配置缓冲池与其他高级打印机选项

对网络连接的打印机，通常需要打印机对文件进行缓存，而不是直接打印。通过打印缓冲池，可以使用打印机（打印队列）对打印任务进行管理。通过Prncnfg与如下一些选项，可以对打印缓冲池进行配置。

- **+Direct**。通过该参数，可以将打印机配置为直接打印，而不需要通过打印缓冲池。并且，指定了该参数后，就不能再使用任意缓冲池选项。
- **-Direct**。通过该参数，对打印文档进行缓存，以便程序可以快速完成打印任务。这是默认的打印选项。
- **+Queued**。通过该参数，在打印文档最后一页缓存之后开始打印。如果希望在打印开始之前缓存整个文档，则可以选定这一选项，确保在打印开始之前整个文档进入打印队列。如果由于某些原因导致打印取消或未完成，则该打印任务不会被打印。
- **-Queued**。通过该参数，只要开始缓存文档，就开始打印。如果希望在打印设备没有被占用的情况下立即开始打印，则可以选定这一选项。如果需要打印任务更快完成，或者需要确保应用程序尽可能快地将控制权转交给用户，则应该选择这一选项，这也是默认选项。
- **+Enabledevq**。通过该参数，缓冲池会对打印机设置进行检查，并在将文档发送到打印设备之前与文档设置进行匹配。如果不匹配，缓冲池会抑制该打印任务，但允许匹配的文档进行打印。如果需要频繁地改变打印机型号或进纸盒分配，则选择这一选项是有益的。
- **-Enabledevq**。通过该参数，缓冲池在将文档发送到打印设备之前不会对打印机设置进行检查。如果不匹配，则打印机通常会终止打印，并等待用户取消该打印任务、改变打印机型号、或插入一个带有合适纸张类型的进纸盒。这是默认的选项。

其他一些可以进行配置的Prncnfg高级选项包括下面列举的。

- **+Keepprintedjobs**。通过该选项，任务在打印后不会从打印队列中删除。如果正在打印的文档不易创建，可以使用这一选项，用于以后需要打印该文档时不必再次创建该文档。
- **-Keepprintedjobs**。通过该选项，任务在打印后会从打印队列中删除，从而释放打印任务占用的磁盘空间，但无法从打印队列中再次打印该文档。这是默认选项。
- **+Docompletefirst**。通过该选项，则完成缓存的任务会比正在缓存的任务优先打印，而不管正在缓存的任务是否具有更高的优先级。这是默认选项。
- **-Docompletefirst**。通过该选项，高优先级的任务将抢占低优先级的任务。设置该选项后，如果有高优先级的任务进入打印队列，则低优先级的任务会终止打印，高优先级的任务将开始打印。
- **+Enablebidi**。通过该参数，可以激活元文件缓存，并打开那些高级打印选项（如果支持），比如打印顺序、书册打印、多页合并打印。使用该选项时，如果注意到兼容性问题，则应该禁

止相关功能。这是默认选项。

- **-Enablebidi**。通过该参数，可以禁用元文件缓存，并关闭那些高级打印选项。如果打印机遇到兼容性问题，则应该使用这一选项。

要了解如何使用这些选项，参考如下一些实例。

将sales06上的SalePrinter配置为直接打印，并保持已打印的任务：

```
prncnfg -t -s sales06 -p "SalesPrinter" +direct +keepprintedjobs
```

将本地计算机上的MainPrinter配置为在最后一页缓存后开始打印：

```
prncnfg -t -p "MainPrinter" -queued
```

将corpsvr09上的HPLaserJet配置为抑制不匹配的文档，并禁用元文件缓存：

```
prncnfg -t -s corpsvr09 -p "HPLaserJet" +enabledevq -enablebidi
```

16.5 解决缓存问题

Windows使用Print Spooler服务控制打印任务的缓存。如果该任务尚未启动，则打印任务无法缓存。

16.5.1 检查 Print Spooler 服务

在本地计算机上，使用如下命令可以检查Print Spooler服务的状态：

```
sc query spooler
```

对远程计算机，则需要指定UNC服务名，比如：

```
sc \\Engsvr04 query spooler
```

无论哪种，输出信息都包含spooler部分，类似于如下格式：

```
SERVICE_NAME: spooler
TYPE          : 110  WIN32_OWN_PROCESS (interactive)
STATE         : 4    RUNNING
               (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
WIN32_EXIT_CODE : 0   (0x0)
SERVICE_EXIT_CODE : 0 (0x0)
CHECKPOINT     : 0x0
WAIT_HINT     : 0x0
```

从上面信息可以看出，Print Spooler服务正在运行。如果该服务终止，则可能需要检查服务配置，为此，可以键入如下命令：

```
sc qc spooler
```

或

```
sc \\Engsvr04 qc spooler
```

输出信息会报告Print Spooler的启动设置，如下所示：

```
[SC] QueryServiceConfig SUCCESS
```



```

SERVICE_NAME: spooler
        TYPE               : 110        WIN32_OWN_PROCESS (interactive)
        START_TYPE          : 2          AUTO_START
        ERROR_CONTROL        : 1          NORMAL
        BINARY_PATH_NAME     : C:\Windows\System32\spoolsv.exe
        LOAD_ORDER_GROUP     : SpoolerGroup
        TAG                  : 0
        DISPLAY_NAME         : Print Spooler
        DEPENDENCIES         : RPCSS
                           : http
        SERVICE_START_NAME   : LocalSystem

```

从上面信息可以看出，启动类型为AUTO_START，表明Print Spooler服务被设置为自动启动。

16.5.2 修复损坏的缓冲池

缓冲池也存在损坏的可能性。如果损坏，则打印机会停止工作，或者不能将打印任务发送到打印设备，有时候还可能导致打印设备打印无法理解的数据。大多数情况下，终止后再重新启动Print Spooler服务就可以解决问题。要终止Print Spooler服务，可以键入如下命令：

```
sc stop spooler
```

缓冲池终止后，如果需要重启该服务，可以键入如下命令：

```
sc start spooler
```

如果操作的是远程计算机，也可以输入远程计算机名，比如：

```
sc \\Engsvr04 stop spooler
sc \\Engsvr04 start spooler
```

注解 有些缓存问题可能与许可权限有关。为解决这些问题，可以在%SystemRoot%\System32\Spool文件夹中检查打印机的访问许可权限与许可权限。

16.6 管理打印队列与单个打印任务

有几个Windows脚本可用于操作打印队列及其包含的打印任务，比如，Prnqctl可用于启动、终止或暂停打印队列中所有文档的打印，Prnjobs可用于操作打印任务。

16.6.1 查看队列中的任务

你可以使用Prnjobs查看打印队列中的任务。如果需要查看本地计算机上所有打印机的所有任务，可以使用命令**prnjobs -l**。要查看特定打印机上的任务，可以使用-P参数指定打印机名。对远程计算机，可以使用-S参数指定待操作的远程计算机。如果必要，还可以使用-U参数与-W参数指定远程登录的用户名与口令。

参考如下实例。

查看CorpSrv03上的所有打印任务：

```
prnjobs -l -s corpsrv03
```

查看本地计算机上MainPrinter的所有打印任务：

```
prnjobs -l -p MainPrinter
```

单个打印任务的输出包含了如下一些信息。

- Job ID。打印任务标识符，在操作单个打印任务时是必需的。
- Printer。打印机名。
- Document。文档文件名，可以包含打印该文档文件的应用程序名。
- Data Type。打印机数据类型。
- Driver Name。打印机驱动程序名，也代表了打印机型号。
- Description。打印机描述信息。
- Elapsed Time。文档打印的时间长度。
- Job Status。打印任务状态，包括正在打印、缓存、暂停、正在删除、正在重启。
- Notify。打印任务完成时应该通知的人（如果已经进行了通知配置）。
- Owner。文档属主。
- Pages Printed。到目前为止已打印的页面数。
- Size。文档大小（以字节计数）。
- Time Submitted。打印任务提交的时间与日期。
- Total Pages。文档的页面总数。

16.6.2 打印机的暂停与恢复

有时候需要暂停打印机，以便操作物理打印设备，对出现的问题进行故障排除。暂停打印机时，打印机完成当前的打印任务，并挂起所有其他打印任务。要暂停打印机，可以使用Prnqctl命令。先键入**prnqctl -z**，之后，对本地打印机，使用**-P**参数设置需要暂停的打印机名。对远程计算机，使用**-S**参数指定待操作的远程计算机。如果必要，还可以使用**-U**参数与**-W**参数指定用户名与口令，以便访问远程计算机。

要恢复暂停的打印机，可以使用**-M**参数（而不是**-Z**参数），用来重启对打印队列中所有文档的打印。

参考如下实例。

暂停CorpSrv03上EngineeringPrinter的打印：

```
prnqctl -z -s corpsrv03 -p EngineeringPrinter
```

暂停本地计算机上5thfloorPrinter上的打印任务：

```
prnqctl -z -p 5thfloorPrinter
```

恢复CorpSrv03上EngineeringPrinter的打印：

```
prnqctl -m -s corpsrv03 -p EngineeringPrinter
```

16.6.3 清空打印队列

你可以使用Prnqctl命令清空打印队列，并删除其中所有内容。先键入**prnqctl -x**，之后，对本地打

印机, 使用-P参数设置需要清空的打印队列名, 对远程计算机, 使用-S参数指定待操作的远程计算机。如果必要, 还可以使用-U参数与-W参数指定用户名与口令, 用来访问远程计算机。

参考如下实例。

清空salespc06上SalesPrinter的打印队列:

```
prnqctl -x -s salespc06 -p SalesPrinter
```

清空本地计算机上TempPrinter上的打印任务:

```
prnqctl -x -p TempPrinter
```

如果命令执行成功, Prnqctl会报告成功地从打印队列中移除了文档, 即便打印队列中本来可能没有文档。

16.6.4 暂停、恢复与重启单个文档的打印

你可以使用Prnjobs暂停或恢复单个文档的打印。暂停某打印任务后, 就挂起了对该任务的打印, 但允许其他文档进行打印。恢复某打印任务后, 就从该任务的挂起点恢复打印。

要暂停某个打印任务, 可以使用如下的语法格式:

```
prnjobs -z -p PrinterName -j JobID
```

其中, *PrintName*是待操作的打印机名, *JobID*是待暂停任务的ID号。

要恢复某个打印任务, 可以使用如下的语法格式:

```
prnjobs -m -p PrinterName -j JobID
```

其中, *PrintName*是待操作的打印机名, *JobID*是待恢复任务的ID号。

上面的两种情况都指的是, 默认情况下, 对本地计算机上的打印机进行操作。对远程计算机上的打印队列, 使用-S参数指定待操作的远程计算机。如果必要, 还可以使用-U参数与-W参数指定用户名与口令, 用来访问远程计算机。

参考如下实例。

暂停CorpSrv03上EngineeringPrinter中编号为6的任务的打印:

```
prnjobs -z -s corpsrv03 -p EngineeringPrinter -j 6
```

暂停本地计算机上5thfloorPrinter上中编号为17的任务的打印:

```
prnjobs -z -p 5thfloorPrinter -j 17
```

恢复CorpSrv03上EngineeringPrinter中编号为6的任务的打印:

```
prnjobs -m -s corpsrv03 -p EngineeringPrinter -j 6
```

成功执行后, Prnjobs会报告成功暂停或恢复某ID打印任务的打印。如果指定的是无效的任务ID, Prnjobs会报告“不能设置打印任务”。

16.6.5 移除文档并取消打印任务

你可以使用Prnjobs取消单独的打印任务并将其从打印队列中删除。先键入prnqctl-x, 之后, 对本地

打印机,使用-P参数设置打印机名,使用-J参数指定待删除文档的ID。对远程计算机,使用-S参数指定待操作的远程计算机。如果必要,还可以使用-U参数与-W参数指定用户名与口令,用于访问远程计算机。

参考如下实例。

取消本地计算机上MainPrinter中ID为12的打印任务:

```
prnjobs -x -p MainPrinter -j 12
```

取消CorpSrv03上EngineeringPrinter中ID为9的打印任务:

```
prnjobs -x -s corpsrv03 -p EngineeringPrinter -j 9
```

成功执行后,Prnjobs会报告成功取消打印任务。如果指定的是无效的任务ID,Prnjobs会报告“不能设置打印任务”。

注解 如果待取消的打印文档正在打印过程中,则打印设备会继续打印部分或全部文档。因为,大多数打印设备会将文档缓存在内部缓冲区中,因而会继续打印缓冲区中的内容。

16.7 备份与恢复打印服务器配置

作为应对故障的常规规划的一部分,应该考虑如何处理打印机与打印服务器的失效。要对潜在的打印机问题有所准备,理想情况下,可以购置空闲的打印机或相应组件,用于修复或替换失效的打印机。如果不能,则应该指定好失效后的策略,以便用户可以使用替代的打印机,或者已经为用户在其计算机上配置好从打印机,以便在主打打印机失效后使用。

16.7.1 备份打印服务器的配置

要对潜在的打印服务器问题进行处理,应该准备一台可用的从打印服务器,或者将某台计算机设置为从打印服务器。作为周期性备份的一部分,你可以使用PrintBrm工具对打印服务器的打印机配置进行备份,使用如下命令:

```
printbrm -b -f SaveFile
```

其中,SaveFile为打印机配置文件(带.printerexport扩展名)的全文件路径。你也可以使用-S参数指定远程计算机。下面的实例中,将打印机配置备份到PrintServer12_Config.printerexport文件中。

```
printbrm -b -f PrintServer12_Config.printerexport
```

在对打印机配置进行备份的过程中,Printbrm工具还会列出打印机当前配置的摘要信息,并按步骤列出任务,如下面实例所示:

```
Operation mode: backup
Target server: local machine
Target file path: c:\Windows\printserver12_config.printerexport.
Queue publish mode: none
Overwrite Mode: keep existing settings
```

```
LISTING PRINT QUEUES
OfficeJetPrinter
HP Color LaserJet 9500 main
```

```

magicolor 2300 DL
Adobe PDF
LISTING PRINTER DRIVERS
hp officejet 5500 series, Windows NT x86, LIDIL hpz111hn
HP Color LaserJet 9500 PS, Windows NT x86, None
magicolor 2300 DL, Windows NT x86, MLMON_B.DLL
Adobe PDF Converter, Windows NT x86, None
LISTING PRINT PROCESSORS
hpzpplhn Windows NT x86 hpzpplhn.dll
MIMFPR_B Windows NT x86 MIMFPR_B.DLL
LISTING PRINTER PORTS
192.168.0.90, TCP
192.168.1.90, TCP
192.168.10.150, TCP
Saving Print Queues...
Saved print queue OfficeJetPrinter
Saved print queue HP Color LaserJet 9500 main
Saved print queue magicolor 2300 DL
Saved print queue Adobe PDF
Saving Print Processors...
Saved print processor hpzpplhn, Windows NT x86, hpzpplhn.dll
Saved print processor MIMFPR_B, Windows NT x86, MIMFPR_B.DLL
Saving Printer Drivers...
Saved printer driver hp officejet 5500 series, Windows NT x86, 3
Saved printer driver HP Color LaserJet 9500 PS, Windows NT x86, 3
Saved printer driver magicolor 2300 DL, Windows NT x86, 3
Saved printer driver Adobe PDF Converter, Windows NT x86, 3
Saving Printer Ports...
Saved printer port 192.168.0.90, TCP
Saved printer port 192.168.1.90, TCP
Saved printer port 192.168.10.150, TCP
***** 100% Complete *****

```

Successfully finished operation.

如果在输出中发现了错误信息，要对导致错误的问题进行解决，之后再次使用Printbrm工具进行备份。要确保打印机配置备份在全服务器故障时是可用的，还应该将配置文件复制到安全的网络文件夹或其他计算机上。

16.7.2 恢复打印服务器的配置

对打印机配置进行备份后，可以有多种方式使用备份文件。发生打印服务器失效后，可以将打印服务器断开网络连接，之后使用Printbrm在新打印服务器上恢复打印机配置，再之后需要改变从打印服务器的IP地址与计算机名，以便与原始打印服务器匹配。设置完毕后，用户就可以访问打印机并恢复打印。

用于恢复已保存打印配置的语法格式如下：

```
printbrm -r -f ConfigFileToRestore
```

其中，*ConfigFileToRestore*为已保存打印机配置文件（带.printerexport扩展名）的全文件路径。下面的实例中，使用PrintServer12_Config.printerexport文件恢复打印机配置：


```
printbrm -r -f PrintServer12_Config.printerexport
```

恢复打印机配置时，Printbrm会列出设置与操作的摘要信息，如下面实例所示：

```
Operation mode: restore
Target server: local machine
Target file path: c:\Windows\printserver12_config.printerexport.
Queue publish mode: none
Overwrite Mode: keep existing settings
Restoring Printer Drivers...
Restoring Printer Ports...
Restoring Print Processors...
Restoring Print Queues...
***** 0% *****
```

Successfully finished operation.

如果输出中包含了错误信息，要先解决导致错误的问题，之后再次运行Printbrm来恢复打印机配置。默认情况下，Printbrm不会重写现存打印队列的设置。在已经包含自己打印队列的计算机上恢复打印队列时，这是有益的。在无意间从打印服务器删除打印队列后，一般希望Printbrm只恢复已删除的打印队列，因而这种不重写设置也是有益的。

要强制Printbrm执行恢复操作并重写现存打印队列的设置，可以使用-O FORCE参数，比如：

```
printbrm -r -f PrintServer12_Config.printerexport -o force
```

如果对已保存打印机配置文件中的内容有疑问，可以使用如下的语法格式查询该文件并列出其中包含的内容：

```
printbrm -q -f ConfigFileToQuery
```

其中，ConfigFileToQuery为已保存打印机配置文件的全文件路径。

16.7.3 迁移打印机与打印队列

你可以使用Printbrm将打印机及其打印队列从打印服务器移动到其他打印服务器。如果需要加固多台打印服务器或替换旧的打印服务器，这是一种有效的方法。

移动打印机时，打印机当前所在服务器为源服务器，打印机将迁移到的服务器为目的服务器。通过如下步骤，可以将打印机移动到新的打印服务器上。

(1) 为源服务器创建打印机配置文件，使用如下命令：**Printbrm -b -s ServerName -f SaveFile**。其中，ServerName为源服务器名或IP地址，SaveFile为.printerexport文件的文件名。

(2) 在目的服务器上恢复打印机配置文件备份，使用如下命令：**Printbrm -r -s ServerName -f RestoreFile**。其中，ServerName为目的服务器名或IP地址，RestoreFile为用于恢复的.printerexport文件的文件名。

如果目的服务器上包含的打印队列与来自源服务器的打印队列同名，可以通过-O FORCE参数使用恢复文件中的打印队列设置重写现存打印队列设置。

传输控制协议/Internet协议（TCP/IP）的配置、维护与故障排除，是每个管理员工作职责的重要组成部分。本章首先对执行这些任务的命令行工具进行了初步讨论，之后细化为每个单独的领域，使读者具备在Windows Vista与Windows Server 2008上成功管理TCP/IP网络的知识与技术。

17.1 使用网络服务 Shell

网络服务shell（Netsh）是一款命令行脚本工具，可用于管理本地与远程计算机上多种网络服务。Netsh提供了一个单独的命令提示符，可以在交互模式或非交互模式下使用。

17.1.1 操作 Netsh 上下文

在交互模式下，键入**netsh**，之后指定待操作网络服务的上下文名称。可用的上下文与子上下文依赖于计算机上已安装的角色服务、角色与功能。关键的上下文名称及其含义如下所示。

- ❑ **advfirewall**。高级防火墙，该上下文用于管理与监控Windows高级安全防火墙。Windows高级安全防火墙是标准Windows防火墙的增强版，可用于定义安全策略，并包含了用于定义高级包过滤规则（包括入站与出站的网络流量）的扩展。
- ❑ **bridge**。网桥，该上下文用于激活或禁用网桥的传输层（OSI模型第3层）兼容模式，也可用于查看网桥的配置设置。
- ❑ **dhcp**。动态主机配置协议（DHCP），该上下文用于查看与管理DHCP服务器。你可以使用DHCP上下文为网络客户端动态地指定TCP/IP配置信息。只有当DHCP Server角色服务已安装后，DHCP上下文在Windows Server 2008上才是可用的。如果尚未安装该角色服务，则上下文**dhcp**的快捷方式将指向**dhcpclient**上下文。
- ❑ **dhcpclient**。DHCP客户端，该上下文用于激活或禁用对DHCP的追踪。
- ❑ **firewall**。Windows防火墙，该上下文用于管理Windows防火墙允许的程序、防火墙端口、日志、通告或其他方面。
- ❑ **http**。超文本传输协议（HTTP），该上下文用于管理HTTP监听器。
- ❑ **interface ip4**。Interface IP版本4（IPv4），该上下文用于查看与管理计算机的IPv4网络配置。要注意的是，很多IPv4 Show命令只有在本地才是可用的。

- ❑ **interface ip6**。Interface IP版本6 (IPv6)，该上下文用于查看与管理计算机的IPv6网络配置。要注意的是，很多IPv6 Show命令只有在本地才是可用的。
- ❑ **interface portproxy**。Interface端口代理，该上下文用于管理IPv4网络、IPv6网络以及两种协议混合网络的代理。
- ❑ **ipsec**。Internet协议安全 (IPsec)，该上下文用于查看与配置IPsec的动态或静态设置。
- ❑ **lan**。有线本地局域网 (LAN)，该上下文用于管理有线网络profiles与操作有线网络接口。大多数lan命令需要利用Wired Autoconfig服务，也就是说，在使用这些命令时，必须先启动该服务。
- ❑ **nap client**。网络访问保护 (NAP) 客户端，该上下文用于管理NAP客户端配置。
- ❑ **nap hra**。NAP 健康注册授权 (HRA)，该上下文用于管理NAP HRA配置。在Windows Server 2008中，只有在安装了健康注册授权角色服务 (作为Network Policy And Access Services角色的一部分) 之后，这一上下文才是可用的。
- ❑ **netio**。网络输入/输出 (netio)，该上下文用于添加、删除与列出网络绑定的过滤器。
- ❑ **ras**。远程访问服务器 (RAS)，该上下文用于查看与管理远程访问服务器的配置。
- ❑ **ras aaaa**。认证、授权、记账与审计 (AAAA)，该上下文用于查看与操作AAAA数据库。Internet 认证服务 (IAS) 以及路由与远程访问服务都使用该数据库。
- ❑ **ras diagnostics**。RAS诊断，该上下文用于配置RAS故障排除过程中的日志与追踪。
- ❑ **routing**。路由，该上下文用于管理路由服务，且与路由与远程访问服务器一起使用。在Windows Server 2008中，只有在安装了Routing And Remote Access Services角色 (作为Network Policy And Access Services角色的一部分) 之后，这一上下文才是可用的。
- ❑ **rpc**。远程过程调用 (RPC) 帮助者，该上下文用于查看与管理计算机上配置的IP地址接口设置与IP子网地址。只有工作在本地时，该上下文中的命令才是可用的。
- ❑ **rpc filter**。RPC防火墙，该上下文用于创建并管理RPC防火墙过滤器及其规则。只有工作在本地时，该上下文中的命令才是可用的。
- ❑ **winhttp**。Windows HTTP (WinHTTP)，该上下文用于管理WinHTTP代理与追踪设置。
- ❑ **wins**。Windows Internet域名服务 (WINS)，该上下文用于查看与管理WINS服务器设置。对Windows以前的系统，可以使用WINS将NetBIOS计算机名解析为IPv4地址。在Windows Server 2008中，只有在安装了WINS Server功能之后，这一上下文才是可用的。
- ❑ **winsock**。Winsock，该上下文用于管理Winsock通信设置。
- ❑ **wlan**。无线LAN，该上下文用于管理无线网络设置。在Windows Server 2008中，只有在安装了Wireless LAN服务之后，这一上下文才是可用的。

注解 从上面的说明中可以看到，有些上下文与命令只有在本地计算机上的Netsh提示符中才是可用的。比如RPC，只有在本地计算机上操作才是可用的。此外，除要求在本地计算机上之外，有些Netsh上下文与命令还要求预先配置了路由与远程访问服务。如果是这种情况，必须设置Connections To Other Access Servers远程访问策略 (以便授予远程访问的许可权限)，并确保远程访问服务已经运行。

在附录B中，可以看到关于Netsh上下文及其子上下文的全面资料。上下文名称用于通知Netsh应该加载哪些起帮助作用的DLL，起帮助作用的DLL提供了可以使用的上下文特定的命令。比如，键入netsh

以便交互式地操作Netsh，之后键入**rpc**，就可以进入RPC上下文，之后键入**show interfaces**，就可以查看计算机上配置的IPv4地址接口。概括地说，上面的实例主要包括如下几个步骤。

- (1) 键入**netsh**，之后命令提示符改变为**netsh>**。
- (2) 键入**rpc**，之后命令提示符改变为**netsh rpc>**。
- (3) 键入**show interfaces**，之后会显示计算机上已配置的IPv4地址接口，如下所示：

Subnet	Interface	Status	Description
127.0.0.0	127.0.0.1	Enabled	Software Loopback Interface 1
192.168.1.0	192.168.1.101	Enabled	Intel(R) PRO/1000 PM
Network Connection			

每个上下文都有不同的命令集，其中的一些命令可能会切换到子上下文，而这些子上下文也可能有自己的命令集。要记住的是，与上下文相关的服务必须已经在计算机或域内进行了配置，这样才能在特定的上下文中进行有意义的操作。不管当前操作的是哪个上下文，都可以键入**help**或**?**来查看可用的命令列表。类似地，不管当前操作的是哪个上下文，都可以键入**exit**或**quit**来退出网络服务shell，并返回到Windows命令提示符。

不管在网络服务shell中的哪个层级，都可以通过键入上下文的完整名来切换到某上下文。比如，如果当前操作的是Interface IPv6上下文，需要切换到Ras Diagnostics上下文，则只需要键入**ras diagnostics**。此外，不管当前操作的是哪个上下文，总是可以使用**..**命令返回到上一层上下文。也就是说，如果当前操作的是Netsh Rpc上下文，但需要切换回顶层的netsh上下文，则可以键入**..**命令，用来返回到上一层上下文。

上面讲解了在交互式模式下使用Netsh。可以看出，交互模式下是比较慢、比较繁重的，但对初学者或者试图发现有哪些可用命令的用户而言，交互式模式是有益的。随着对Netsh的熟悉，你可能需要以非交互模式使用Netsh。在非交互模式下，可以在命令提示符中或脚本中键入完整的命令序列。比如，前面讲的实例中使用了3个步骤，实际上可以在同一条命令中执行：

```
netsh rpc show interfaces
```

不管是在命令行中执行上面的命令，还是在脚本中执行，其输出都是一样的，即当前操作的计算机上的网络接口列表。可以看出，直接键入完整的命令序列要快得多。

17.1.2 操作远程计算机

你可以使用Netsh来操作远程计算机。如果需要交互式地操作远程计算机，可以使用**-R**参数，其后跟随远程计算机的IP地址或域名，比如：

```
netsh -r 192.168.10.15
```

或

```
netsh -r corpsvr02
```

操作远程计算机时，Netsh会在命令提示符中包含该计算机的IP地址或计算机名，比如：

```
[corpsvr02] netsh>
```

上面的提示符表明，当前正在使用Netsh操作远程计算机CorpSvr02。通过如下的参数，可以指定必要的用户登录凭据。

□ **-u:**。指定不同的用户（以Domain\User或User的形式）登录到远程计算机，且只有在操作远程计算机时才是可用的。

□ **-p:**。为指定的用设置口令。如果不设置口令，或使用 "*" 作为口令，则会看到提示符要求输入口令。操作远程计算机时，只有在指定了不同的登录用户时才会使用这一参数。

下面的实例中，使用账号CPANDL\WilliamS登录远程计算机FileServer25，并操作其上的Netsh:

```
netsh -r fileserver25 -u cpandl\williams -p *
```

如果需要以非交互模式操作远程计算机，则必须使用如下的语法格式:

```
netsh -c Context -r RemoteComputer Command
```

其中，*Context*为待操作上下文的标识符，*RemoteComputer*为远程计算机的计算机名或IP地址，*Command*为待执行的命令。参考如下实例:

```
netsh -c "interface ipv4" -r corpsvr02 show interfaces
```

上面的实例中，使用Interface IPv4这一上下文获取CorpSvr02上已配置的网络接口列表。要注意的是，这里不能使用RPC上下文完成这一任务，因为该上下文只能在本地计算机上使用。

真实场景 要使用Netsh与远程计算机交互，必须先在网上配置路由与远程访问服务。特别地，必须设置Connections To Other Access Servers远程访问策略（以便授予远程访问的许可权限），并确保远程访问服务已经运行。

17.1.3 操作脚本文件

前面提到过，你可以在命令行中输入完整的Netsh命令序列，也可以在批处理脚本中输入，但必须熟练掌握要输入的完整命令，而不能依赖于Netsh的帮助。有些命令行可能是非常长而复杂的，比如，如下的命令连接到DHCP服务器、配置DHCP范围并激活该范围:

```
netsh dhcp server \\corpsvr02 add scope 192.168.1.0 255.255.255.0 MainScope  
PrimaryScope
```

```
netsh dhcp server \\corpsvr02 scope 192.168.1.0 add iprange 192.168.1.1  
192.168.1.254
```

```
netsh dhcp server \\corpsvr02 scope 192.168.1.0 add excluderange 192.168.  
1.1 192.168.1.25
```

```
netsh dhcp server \\corpsvr02 scope 192.168.1.0 set state 1
```

将上面的命令保存到批处理脚本中之后，就可以像运行其他脚本一样运行该脚本。比如，如果将该脚本命名为dhcpconfig.bat，则可以在命令行中键入**dhcpconfig**来运行该脚本。

操作远程计算机时，可以将该脚本存储在远程计算机可以访问的网络共享位置，之后远程登录并运行该脚本。也可以将脚本直接复制到远程计算机，之后远程登录计算机并执行该脚本。两种方式都可以正常工作，但都涉及到一些附加的步骤。幸运的是，还有一种更快捷地在远程计算机上运行脚本的方法。为此，必须对脚本做一些修改，并使用如下的语法格式:

```
netsh -c Context -r RemoteComputer -f Script
```

其中，*Context*为待操作上下文的标识符，*RemoteComputer*为远程计算机的计算机名或IP地址，*Script*

为待执行脚本的文件或网络路径。参考如下实例：

```
netsh -c "dhcp server" -r corpsvr02 -f dhcpconfig.bat
```

上面的实例中，在CorpSvr02上使用DHCP Server上下文运行了一个名为dhcpconfig.bat的脚本。注意的是，Server上下文是DHCP上下文的子上下文，该脚本中包含如下一些命令：

```
add scope 192.168.1.0 255.255.255.0 MainScope PrimaryScope
scope 192.168.1.0 add iprange 192.168.1.1 192.168.1.254
scope 192.168.1.0 add excluderange 192.168.1.1 192.168.1.25
scope 192.168.1.0 set state 1
```

这些命令创建、配置并激活了指定的DHCP Server (CorpSvr02) 上的DHCP范围。由于已经处于CorpSvr02上的DHCP Server上下文中，因此不需要在每条命令前键入netsh dhcp server \corpsvr02。

17.2 管理 TCP/IP 设置

计算机使用IP地址并通过TCP/IP协议进行通信。Windows Vista与Windows Server 2008实现了双IP层体系结构，同时支持IPv4与IPv6，并共享通用的传输层与数据帧层。IPv4与IPv6在设计与实现上存在很大的差别。IPv4使用32位的地址格式，是大多数网络上使用的主要IP版本；IPv6使用128位的地址格式，是下一代IP版本。

安装操作系统时，检测到网卡等网络硬件后，IPv4与IPv6都会默认激活，而不需要安装单独的组件来激活对IPv6的支持。IP地址可以在命令行中手工或动态配置。手工配置时，会为计算机指定一个静态的IP地址。静态地址是固定的，除非手工改变，否则不会改变。动态配置时，计算机会从网络上的DHCP服务器动态获取一个IP地址，该地址只有在计算机启动后才会从DHCP服务器获取，并可能每次获取不同的IP地址。典型情况下，在域中，Windows服务器使用静态IP地址，Windows工作站则使用动态IP地址。

17.2.1 配置 IPv4

IPv4使用32位的地址格式，通常表示为4个单独的十进制数值，比如127.0.0.1或192.168.10.50。这4个十进制数值也称为八位位组，因为每个十进制数值实际上代表了32位IP地址中的8位（由于8位的限制，十进制数值的范围限定为0到255）。对标准的单播IPv4地址，一部分代表的是网络ID，另一部分代表的是主机ID。要注意的是，主机的IPv4地址与主机的网络适配器使用的内部机器地址（MAC地址）并没有必然的关联。

1. 设置静态IPv4地址

设置IPv4地址时，实际上就是为计算机指定要使用的IPv4地址、该地址的掩码以及默认的网关（如果必要）。完成这些配置后，可能也需要为域名系统（DNS）以及WINS进行域名解析配置。

要指定IPv4地址，需要使用Interface IPv4上下文。命令为SET ADDRESS，语法格式如下：

```
set address [name=]InterfaceName source=static address=IPAddress
mask=SubnetMask [gateway={none | DefaultGateway
[[gwmetric=]GatewayMetric]]
```

注解 如果计算机在指定的网络接口上已经配置了IPv4地址，则使用SET ADDRESS命令会替代原来设置的值。要为该网络接口添加地址而不是替换原来的地址，可以使用ADD ADDRESS命令。

在命令行中键入 **netsh interface ipv4 show addresses**, 或者在 Netsh Interface IPv4 上下文中键入 **show addresses**, 可以查看可用的网络接口及其当前配置。如下面实例所示, 输出信息指定了可用的网络接口名及其当前配置:

```
Configuration for interface "Local Area Connection"
    DHCP enabled :           Yes
    IP Address:              192.168.1.101
    Subnet Prefix:           192.168.1.0/24 (mask 255.255.255.0)
    Default Gateway:         192.168.1.1
    Gateway Metric:          0
    InterfaceMetric:         10

Configuration for interface "Loopback Pseudo-Interface 1"
    DHCP enabled:            No
    IP Address:              127.0.0.1
    Subnet Prefix:           127.0.0.0/8 (mask 255.0.0.0)
    InterfaceMetric:         50
```

大多数情况下, 需要操作的网络接口名为“本地区域连接”。上面的实例中, 列出的伪接口用于本地回环通信。要注意的是, 为计算机指定的IPv4地址必须尚未在网络上使用。子网掩码字段用于确保计算机在网络上的正确通信。如果网络使用掩码, 则公司内每个网段使用的网络掩码都是不同的。如果计算机需要访问其他TCP/IP网络、Internet或其他子网, 还必须指定默认的网关, 网关可以使用网络的默认路由器的IPv4地址。

网关metric指明了使用网关的相对路由成本。如果某个特定的IPv4地址有多个可用的默认路由, 则路由成本最低的网关最先使用。如果该网关无法使用, Windows会尝试使用路由成本次低的网关, 并依此类推。与GUI中不同的是, 命令行中不会自动为网络指定metric, 而必须对其进行手工指定。参考如下实例:

```
set address name="Local Area Connection" source=static
address=192.168.1.50 mask=255.255.255.0 gateway=192.168.1.1
gwmetric=1
```

上面的实例中, 操作的网络接口为“本地区域连接”, 设置的静态IPv4地址为192.168.1.50, 网络掩码为255.255.255.0, 默认网关为192.168.1.1, 网关metric为1。

提示 在命令行中键入 **netsh interface ipv4 show addresses**, 或者在 Netsh Interface IPv4 上下文中键入 **show addresses**, 可以查看并确认刚才所做的配置。由于很多 Interface IPv4 Show 命令与 Interface IPv6 Show 命令只有在本地工作时才是可用的, 因此要使用这些命令, 必须首先本地登录。

2. 设置动态IPv4地址

只要网络上的DHCP服务器是可用的, 就可以为计算机上的任意网络适配器分配动态的IPv4地址, 之后依赖于DHCP服务器来提供必要的IPv4地址信息。由于动态IPv4地址是变化的, 因此通常不应该为Windows Server 2008等服务器分配动态IPv4地址。

在 Netsh Interface IPv4 上下文中, 使用 SET ADDRESS 命令, 可以设置动态IPv4地址, 其语法格式如下:

```
set address name=InterfaceName source=dhcp
```

参考如下实例：

```
set address name="Local Area Connection" source=dhcp
```

上面的实例中，工作在Netsh Interface IPv4上下文中，并为“本地区域连接”这一网络接口设置动态IPv4地址。

3. 添加IPv4地址与网关

Windows Vista与Windows Server 2008中，都可以使用多个IPv4地址，即便计算机可能只有一个网络适配器。如果需要单一计算机充当几台计算机，或者所在网络划分为几个不同子网，而计算机需要访问这些子网来提供路由信息（或提供其他网络服务时），指定多个IPv4地址是有用的。

注解 要记住的是，使用单一的网络适配器时，多个IPv4地址必须属于同一网段或同一逻辑网的不同网段。如果网络包含了不同的物理网络，则必须使用多个网络适配器，并且每个适配器分配一个属于不同物理网段的IPv4地址。

在Netsh Interface IPv4上下文中，使用ADD ADDRESS命令，可以为单一网络适配器指定多个IPv4地址与网关，其语法格式与SET ADDRESS类似：

```
add address [name=]InterfaceName address=IPAddress mask=SubnetMask
[[gateway=]DefaultGateway [gwmetric=]GatewayMetric]
```

参考如下实例：

```
add address name="Local Area Connection" address=192.168.2.12
mask=255.255.255.0 gateway=192.168.2.1 gwmetric=1
```

注解 如果指定网关，就必须同时指定网关metric。与前面一样，可以通过show addresses命令查看所做的配置。

上面的实例中，操作的网络接口为“本地区域连接”，设置的静态IPv4地址为192.168.2.12，网络掩码为255.255.255.0，默认网关为192.168.2.1，网关metric为1。

4. 设置IPv4地址使用的DNS服务器

计算机使用DNS来完成主机名与IP地址之间的解析。对使用静态IPv4地址的计算机，必须为其指定使用哪一台DNS服务器，这可以在Netsh Interface IPv4上下文中指定，其语法格式如下：

```
set dnsserver name=InterfaceName source=static address=DNSAddress
```

参考如下实例：

```
set dnsserver name="Local Area Connection" source=static
address=192.168.1.56
```

上面的实例中，操作的网络接口为“本地区域连接”，并指定DNS服务器的IP地址为192.168.1.56。配置静态的DNS服务器地址时，可以使用可选的register参数来控制DNS的登记，要记住如下几点。

- 默认情况下，网络接口的所有IP地址都在计算机完全限定域名下的DNS中登记。这种自动登记使用了DNS动态更新协议。如果需要禁用，可以使用命令register=none。
- 默认情况下，计算机的全名只在其主域内登记，因为默认情况下的设置是register=primary。使用动态DNS时，也可以指定连接特定的DNS名应该在DNS中进行登记。为此，要使用命令

register=both。在计算机有多个网络适配器并连接到多个域时，这种做法是有益的。

如果计算机使用DHCPv4，并希望由DHCPv4提供DNS服务器地址，则可以提供DNS服务器地址，或指定从DHCPv4获取IPv4地址——这是通过如下命令实现的：

```
set dnsserver name=InterfaceName source=dhcp
```

参考如下实例：

```
set dnsserver name="Local Area Connection" source=dhcp
```

上面的实例中，指定网络接口“本地区域连接”应该从DHCPv4获取DNS服务器地址设置。

注解 如果计算机已经设置了DNS服务器的IPv4地址，则可以通过SET DNSSERVER命令替换现存值。如果希望添加DNS服务器的IPv4地址，而不是替换现存值，则可以使用命令ADD DNSSERVER。要确认DNS服务器的设置，可以键入命令**show dnsservers**。

5. 指定附加的DNS服务器

大多数网络都有多台DNS服务器来提供域名解析服务，这样才可以保证在某台DNS服务器失效的时候由其他DNS服务器继续提供服务。使用DHCPv4指定DNS服务器时，会自动地通知计算机还有哪些可用的DNS服务器，但手工指定DNS服务器时则并非如此。

要通知计算机还有哪些可用的DNS服务器（除主DNS服务器之外），可以在Netsh Interface IPv4上下文中使用ADD DNSSERVER命令，其语法格式如下：

```
add dnsserver name=InterfaceName address=DNSAddress
```

参考如下实例：

```
add dnsserver name="Local Area Connection" address=192.168.1.75
```

上面的实例中，操作的网络接口为“本地区域连接”，并指定一个备用的DNS服务器，其IP地址为192.168.1.75。

默认情况下，新添加的DNS服务器会添加到TCP/IP配置中DNS服务器列表的最后。如果需要其出现在DNS服务器列表中的特定位置，可以使用参数Index=。比如，如果需要附加的服务器出现在列表首部（即作为主DNS服务器），则应该将其索引设置为1，比如：

```
add dnsserver name="Local Area Connection" address=192.168.1.75 index=1
```

6. 设置WINS服务器

WINS用于将NetBIOS名解析为IP地址，计算机可以通过WINS确定网络上Windows 2000以前计算机的地址。尽管所有版本的Windows都支持WINS，但Windows Server 2008支持WINS主要是为了保证向后兼容。

对使用静态IP地址的计算机，必须指定要使用的WINS服务器。在Netsh Interface IPv4上下文中，用于指定特定WINS服务器的语法格式如下：

```
set winsserver name=InterfaceName source=static address=WINSAddress
```

参考如下实例：

```
set winsserver name="Local Area Connection" source=static  
address=192.168.1.64
```

上面的实例中，操作的网络接口为“本地区域连接”，并将WINS服务器的IP地址指定为192.168.1.64。

如果计算机使用DHCP，并希望由DHCP提供DNS服务器地址，则可以提供WINS服务器地址，或指定从DHCP获取WINS地址——这是通过如下命令实现的：

```
set winsserver name=InterfaceName source=dhcp
```

参考如下实例：

```
set winsserver name="Local Area Connection" source=dhcp
```

上面的实例中，指定网络接口“本地区域连接”应该从DHCP获取WINS服务器地址设置。

注解 如果计算机已经设置了WINS服务器的IP地址，则SET WINSSERVER命令会替换现存值。如果希望添加WINS服务器的IP地址，而不是替换现存值，则可以使用命令ADD WINSSERVER。要确认WINS服务器的设置，可以键入命令**show winsservers**。

7. 指定附加的WINS服务器

大多数网络都有一台主WINS服务器与一台备份WINS服务器，这样才可以保证在某台WINS服务器失效的时候由另外的WINS服务器继续提供服务。使用DHCP指定WINS服务器时，会自动地通知计算机还有哪些可用的WINS服务器，但手工指定WINS服务器时则并非如此。

要通知计算机还有哪些可用的WINS服务器（除主WINS服务器之外），可以在Netsh Interface IPv4上下文中使用ADD WINSSERVER命令，其语法格式如下：

```
add winsserver name=InterfaceName address=WINSAddress
```

参考如下实例：

```
add winsserver name="Local Area Connection" address=192.168.1.155
```

上面的实例中，操作的网络接口为“本地区域连接”，并指定一个备用的WINS服务器，其IP地址为192.168.1.155。

默认情况下，新添加的WINS服务器会添加到TCP/IP配置中WINS服务器列表的最后。如果需要其出现在WINS服务器列表中的特定位置，可以使用参数Index=。比如，如果需要附加的服务器出现在列表首部（即作为主WINS服务器），则应该将其索引设置为1，比如：

```
add winsserver name="Local Area Connection" address=192.168.1.155 index=1
```

8. 删除IPv4地址解析协议缓存

计算机查找IPv4地址的域名信息时，相关的信息会存储在地址解析协议（ARP）缓存中，以便在下次需要该信息时不需要再次查询。地址解析信息会根据接受该信息时的生存时间（TTL）设置来确定自身的过期期限，过期后，必须再次进行查询以便获取当前信息与新的TTL。通常，域名信息的这种自动化获取、清除、更新机制可以正常运作。然而，有时候，系统中的陈旧域名解析信息会在其清除之前造成问题。比如，如果计算机改变了自身的域名信息，但前次查询获取的域名信息与TTL尚未过期，则可能会导致暂时无法查找该计算机的情况。

DNS管理员有几个技巧可用于降低域名信息改变造成的不利影响，比如将TTL设置为逐渐变短，使其恰好赶在域名改变之前过期，以便确保陈旧信息尽快删除，防止出现问题。然而，更方便的方法实际上就是清除陈旧信息，并强制计算机进行新的DNS查询。为此，可以在命令行中键入**netsh interface**

ipv4 delete arpcache, 或者在Netsh Interface IPv4上下文中键入**delete arpcache**, 这一命令将删除当前操作计算机上所有已配置的网络接口的名信息。如果计算机上有多个网络接口, 而只需要清除某个接口上的名信息, 则可以通过参数**name=InterfaceName**指定具体的网络接口, 比如:

```
delete arpcache name="Local Area Connection"
```

9. 删除TCP/IPv4设置

使用Netsh Interface IPv4上下文, 也可以删除TCP/IPv4配置信息, 表17-1根据待执行的具体任务分别总结了可用的命令。

表17-1 用于删除TCP/IPv4配置的Netsh Interface IPv4命令

任 务	语 法	实 例
从指定的网络接口中删除指定的IP地址	<code>delete address name=(InterfaceName) address=(IPAddress)</code>	<code>delete address name="Local Area Network" address=192.168.1.5 6</code>
从指定的网络接口中删除一个静态的网关IPv4地址	<code>delete address) name=(InterfaceName) gateway=(GatewayAddress)</code>	<code>delete address name="Local Area Network" gateway=192.168.1.1</code>
从指定的网络接口中删除所有静态的网关IPv4地址	<code>delete address) name=(InterfaceName) gateway=(all)</code>	<code>delete address name="Local Area Network" gateway=all</code>
从指定的网络接口中删除一台DNS服务器	<code>delete dnsserver) name=(InterfaceName) address=(IPAddress)</code>	<code>delete dnsserver name="Local Area Network" address=192.168.1.5 6</code>
从指定的网络接口中删除所有DNS服务器	<code>delete dnsserver) name=(InterfaceName) address=(all)</code>	<code>delete dnsserver name="Local Area Network" address=all</code>
从指定的网络接口中删除一台WINS服务器	<code>delete winsserver) name=(InterfaceName) address=(IPAddress)</code>	<code>delete winsserver name="Local Area Network" address=192.168.1.5 6</code>
从指定的网络接口中删除所有WINS服务器	<code>delete winsserver) name=(InterfaceName) address=(all)</code>	<code>delete winsserver name="Local Area Network" address=all</code>

17.2.2 配置 IPv6

IPv6使用的是128位地址空间, 128位地址被划分为8个16位地址块, 之间由冒号分隔, 每个地址块都由16进制表示。标准的单播IPv6地址的128位中, 前64位代表的是网络ID, 后面64位代表的是网络接口。比如, FEC0:0:0:02BC:FF:BECB:FE4F:961D。由于IPv6地址的很多地址块都设置为0, 临近的0地址块可以表示为“::”, 在实际的地址中, 并不使用这个双引号。上面给出的地址实例中, 连续的两个0地址块就可以这样表示, 压缩后的地址格式为FEC0::02BC:FF:BECB:FE4F:961D。3个或更多连续的0地址块也可以类似处理, 比如, FFE8:0:0:0:0:0:0:1可以压缩为FFE8::1。

1. 设置IPv6地址

默认情况下, 计算机已经自动设置了IPv6地址。使用IPv6的计算机连接到网络时, 会发送一个链路本地多播请求, 用来取回配置设置。使用IPv6的计算机也可以使用DHCPv6从DHCPv6服务器获取IPv6配置信息。在网络上配置DHCPv6服务器时, 要指定DHCPv6如何为客户端提供服务, 但并不需要修改动态的客户端配置。典型情况下, 自动分配的IPv6地址是链路本地地址, 只能在本地上网访问。

对需要可路由IPv6地址的计算机, 需要为其指定静态的IPv6地址。设置静态IPv6地址时, 还需要指定该地址是单播还是anycast。默认情况下为单播地址, 主要有唯一本地单播与全局单播两种单播地

址。唯一本地单播IPv6地址可以在内部网络上进行路由，但Internet上不可以访问。全局单播IPv6地址则可以在Internet上进行路由，比如为外部服务器分配这种地址。Anycast地址可以分配给多个网络接口，比如为计算机上所有网络接口分配同一个IPv6地址。

要指定静态IPv6地址，需要使用Netsh Interface IPv6上下文，命令为SET ADDRESS，语法格式如下：

```
set address [interface=]InterfaceName address=IPAddress
type=AddressType
```

注解 如果计算机在指定的网络接口上已经配置了IPv6地址，则使用SET ADDRESS命令会替代原来设置的值。要为该网络接口添加地址而不是替换原来的地址，可以使用ADD ADDRESS命令。

在命令行中键入**netsh interface ipv6 show addresses**，或者在Netsh Interface IPv6上下文中键入**show addresses**，可以查看可用的网络接口及其当前配置。如下面实例所示，输出信息指定了可用的网络接口名及其当前配置：

Interface 1: Loopback Pseudo-Interface 1

Addr Type	DAD State	Valid Life	Pref. Life	Address
Other	Preferred	infinite	infinite	::1

Other	Preferred	infinite	infinite	::1
-------	-----------	----------	----------	-----

Interface 7: Local Area Connection

Addr Type	DAD State	Valid Life	Pref. Life	Address
Other	Preferred	infinite	infinite	fe80::6712:1345:cc87:3820%7

Other	Preferred	infinite	infinite	fe80::6712:1345:cc87:3820%7
-------	-----------	----------	----------	-----------------------------

Interface 11: Local Area Connection* 8

Addr Type	DAD State	Valid Life	Pref. Life	Address
Other	Preferred	infinite	infinite	fe80::5efe:192.168.1.101%11

Other	Preferred	infinite	infinite	fe80::5efe:192.168.1.101%11
-------	-----------	----------	----------	-----------------------------

注解 支持IPv6的计算机将fe80::/64 link-local网络地址用于连接到网络(但没有IPv6路由器或DHCPv6服务器)的网络适配器。如果计算机上配置了多个网络接口，则会在IP地址后跟随一个百分号与数值。比如，%7会添加到Interface 7的输出中，%11会添加到Interface 11的输出中。

参考如下实例：

```
set address interface="Local Area Connection"
address= 2001:1cb7::2b58:02bb:00ff:fe45:bc7d type= unicast
```

上面的实例中，操作的网络接口为“本地区域连接”，设置的静态单播IPv6地址为2001:1cb7::2b58:02bb:00ff:fe45:bc7d。

提示 在命令行中键入**netsh interface ipv6 show addresses**，或者在Netsh Interface IPv6上下文中键入**show addresses**，可以查看并确认刚才所做的配置。由于很多Interface IPv6与Interface IPv6 Show命令只有在本地工作时才是可用的，因此要使用这些命令，必须首先本地登录。

在Netsh Interface IPv6上下文中，使用ADD ADDRESS命令，可以为单一网络适配器指定多个IPv6地址，其语法格式类似于SET ADDRESS，如下所示：

```
add address [interface=]InterfaceName address=IPAddress
type=AddressType
```

2. 设置IPv6地址使用的DNS服务器

对使用静态IPv6地址的计算机，必须为其指定使用哪一个DNS服务器。这可以在Netsh Interface IPv6上下文中指定，其语法格式如下：

```
set dnsserver name=InterfaceName source=static address=DNSAddress
register=
```

参考如下实例：

```
set dnsserver name="Local Area Connection" source=static
address=fec0:0:0:ffff::1
```

上面的实例中，操作的网络接口为“本地区域连接”，并指定DNS服务器的IP地址为fec0:0:0:ffff::1。

配置静态的DNS服务器地址时，可以使用可选的register参数来控制DNS的登记。使用命令register=none可以禁用DNS的动态更新，使用命令register=primary可以将计算机的完整名在主域内登记（默认设置），使用命令register=both可以指定connection-specific DNS名应该以DNS以及主DNS的后缀登记。

如果计算机使用DHCPv6，并希望由DHCPv6提供DNS服务器地址，则可以提供DNS服务器地址，或指定从DHCPv6获取IPv6地址。要使得计算机从DHCPv6获取DNS服务器地址，可以使用如下命令：

```
set dnsserver name=InterfaceName source=dhcp
```

参考如下实例：

```
set dnsserver name="Local Area Connection" source=dhcp
```

上面的实例中，指定网络接口“本地区域连接”应该从DHCP获取DNS服务器地址设置。

注解 如果计算机已经设置了DNS服务器的IPv6地址，则SET DNSSERVER命令会替换现存值。如果希望添加DNS服务器的IP地址，而不是替换现存值，则可以使用命令ADD DNSSERVER。要确认DNS服务器的设置，可以键入命令show dnsservers。

要通知计算机还有哪些可用的DNS服务器（除主DNS服务器之外），可以在Netsh Interface IPv6上下文中使用ADD DNSSERVER命令，其语法格式如下：

```
add dnsserver name=InterfaceName address=DNSAddress
```

参考如下实例：

```
add dnsserver name="Local Area Connection" address=fec0:0:0:ffff::2
```

上面的实例中，操作的网络接口为“本地区域连接”，并指定一台备用的DNS服务器，其IP地址为fec0:0:0:ffff::2。

默认情况下，新添加的DNS服务器会添加到TCP/IP配置中DNS服务器列表的最后。如果需要其出现在DNS服务器列表中的特定位置，可以使用参数Index=。比如，如果需要附加的服务器出现在列表首部（即作为主DNS服务器），则应该将其索引设置为1，比如：

```
add dnsserver name="Local Area Connection" address= fec0:0:0:ffff::2
index=1
```

3. 删除TCP/IPv6设置

使用Netsh Interface IPv6上下文，也可以删除TCP/IPv6配置信息，表17-2根据待执行的具体任务分别总结了可用的命令。

表17-2 用于删除TCP/IPv6配置的Netsh Interface IPv6命令

任 务	语 法	实 例
从指定的网络接口中删除一个指定的IP地址	<code>delete address name=InterfaceName address=IPAddress</code>	<code>delete address name="Local Area Network" address=2001:1cb7::2b58:02bb:00ff:fe45:bc7d</code>
从指定的网络接口中删除一台DNS服务器	<code>delete dns name=InterfaceName address=IPAddress</code>	<code>delete dns name="Local Area Network" address=fec0:0:0:ffff::2</code>
从指定的网络接口中删除所有DNS服务器	<code>delete dns name=InterfaceName address=all</code>	<code>delete dns name="Local Area Network" address=all</code>

17.3 支持 TCP/IP 网络

Netsh shell提供了两种用于操作TCP/IP的上下文：Interface IPv4上下文用于查看TCP/IPv4的统计资料并改变设置，Interface IPv6上下文用于查看TCP/IPv6的统计资料并改变设置。使用这些上下文的前提是计算机中已经安装了必要的TCP/IP网络组件，如果尚未安装，则需要先安装再操作。

17.3.1 获取并保存 TCP/IP 设置

如果用过Windows，你可能已经知道，在命令提示符中键入**ipconfig**，就可以获取IPv4与IPv6的基本配置信息，比如：

```
Windows IP Configuration
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::6712:1345:cc87:3820%7
    IPv4 Address. . . . . : 192.168.1.101
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Tunnel adapter Local Area Connection* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Tunnel adapter Local Area Connection* 8:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::5efe:192.168.1.101%11
    Default Gateway . . . . . :
```

从上面可以看出，输出信息中包含了IPv6链路本地地址以及IPv4地址、子网掩码、以及本地局域网连接以太网适配器的默认网关。如果需要获取更详尽的资料，可以使用命令**ipconfig /all**显示一些附加的信息，包括适配器的物理地址（MAC地址）、DHCP状态、使用的DNS服务器以及主机信息，比如：

Windows IP Configuration

```

Host Name:      salespc09
Primary Dns Suffix:  cpandl.com
Node Type:      Hybrid
IP Routing Enabled:  No
WINS Proxy Enabled:  No
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix  . :
    Description . . . . . : Intel(R) PRO/
1000 PM Network Connection
    Physical Address . . . . . : EA-BF-C2-D4-EF-12
    DHCP Enabled . . . . . : Yes
    Autoconfiguration Enabled . . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::6712:1345:cc87:3820%7 (Preferred)
    IPv4 Address . . . . . : 192.168.1.35 (Preferred)
    Subnet Mask . . . . . : 255.255.255.0
    Lease Obtained . . . . . : Friday, April 04, 2009 9:37:43 AM
    Lease Expires . . . . . : Saturday, April 05, 2009 12:05:32 PM
    Default Gateway . . . . . : 192.168.1.1
    DHCP Server . . . . . : 192.168.1.50
    DHCPv6 IAID . . . . . : 128384737
    NetBIOS over Tcpip . . . . . : Enabled

```

从这些信息可以看出,完全限定域名为salespc09.cpandl.com配置为使用DHCP,IP地址为192.168.1.35,子网掩码255.255.255.0。由于IP地址是动态分配的,因此包含特定的Lease Obtained 与Lease Expiration日期与时间戳。

如果在命令提示符中键入**netsh interface ipv4 show config**,可以获取类似的、缩略的IPv4配置信息,比如:

```

Configuration for interface "Local Area Connection"
    DHCP enabled :      No
    IP Address:      192.168.1.50
    Subnet Prefix:    192.168.1.0/24 (mask 255.255.255.0)
    Default Gateway:  192.168.1.50
    GatewayMetric:    256
    InterfaceMetric:  20
    Statically Configured DNS Servers:  127.0.0.1
    Register with which suffix:  Primary only
    Statically Configured WINS Servers:  None

```

通过键入**netsh interface ipv6 show addresses**以及**netsh interface ipv6 show dnsservers**,可以获取基本的IPv6配置信息。上面给出的是获取TCP/IP配置信息的所有方法。

Netsh还提供了保存IPv4、IPv6配置信息的方法,这样只通过运行Netsh脚本就可以重建这些设置。如果要将IPv4配置信息保存到一个文件,可以键入如下命令:

```
netsh interface ipv4 dump > FileName
```

其中,FileName为待保存IPv4配置信息的文件名,命令清单17-1展示了一个IPv4配置文件实例:

命令清单17-1 IPv4配置脚本

```
# -----
```



```
# IPv4 Configuration
# -----
pushd interface ipv4

set address name="Local Area Connection" source=static
address=192.168.1.50 mask=255.255.255.0
set address name="Local Area Connection" gateway=192.168.1.1
gwmetric=1
set dnsserver name="Local Area Connection" source=static address=192.
168.1.56 register=primary
set winsserver name="Local Area Connection" source=static address=none

popd
# End of interface IPv4 configuration
```

命令清单17-1是一个Netsh脚本，可以通过如下的语法格式运行：

```
netsh -c "interface ipv4" -f FileName
```

参考如下实例：

```
netsh -c "interface ipv4" -f corpsvr02-ipconfig.txt
```

这一实例中，使用Interface IPv4上下文运行一个名为corpsvr02-ipconfig.txt的Netsh脚本，以便应用其中定义的IPv4配置。创建配置脚本的关键原因是可以保存IP配置的备份。如果配置无意间被错误修改，就可以从脚本中恢复原始配置。

17.3.2 检查 IP 地址与网络接口配置

Netsh Interface IPv4与Netsh Interface IPv6上下文提供了几条命令，可用于查看IP地址与网络接口的配置信息。这里，网络接口是指计算机用于TCP/IP的某个网络适配器。大多数计算机都有两个网络接口：一个本地回环网络接口，一个本地区域连接网络接口。

对IPv4，本地回环接口是一个伪接口，使用的IPv4地址为127.0.0.1，网络掩码为255.255.255.0。所有通过这一接口发送的IPv4信息都会回环到该计算机本身，而不会发送到网络上。

对IPv6，本地回环接口的IPv6地址为::1。所有通过这一接口发送的IPv6信息都会回环到该计算机本身，而不会发送到网络上。

安装TCP/IP网络时，本地区域连接网络接口是自动创建的，每个网络适配器都有这样一个接口。默认情况下，第一个接口命名为本地区域连接，第二个定义为本地区域连接2，依此类推。

在Windows命令行中，键入**netsh interface ipv4 show global**，可以查看IPv4的全局配置信息，其输出应该类似于如下格式：

General Global Parameters

```
-----
Default Hop Limit           : 128 hops
Neighbor Cache Limit        : 256 entries per interface
Route Cache Limit           : 128 entries per compartment
Reassembly Limit            : 27229728 bytes
ICMP Redirects              : enabled
Source Routing Behavior     : dontforward
Task Offload                : enabled
Dhcp Media Sense            : enabled
```

```

Media Sense Logging      : enabled
MLD Level                : all
MLD Version              : version3
Multicast Forwarding     : disabled
Group Forwarded Fragments : disabled
Randomize Identifiers    : enabled
Address Mask Reply       : disabled
Current Global Statistics
-----
Number of Compartments   : 1
Number of NL clients     : 7
Number of FL providers   : 4

```

上面的信息列出了计算机上配置的所有网络接口IPv4全局状态。默认的Hop Limit列出了数据包在网络上进行路由时的最大默认跳数，Reassembly Limit指定了IP数据报的最大重组尺寸。这一实例中，使用这一网络接口接收或发送的IP数据报最大重组尺寸为27,229,728字节，但实际上数据块一般不会以这个大小发送，而是进行分片发送，最后再重组为一个完整的IP数据包。稍后会详细讨论IP数据报分片问题。

使用**netsh interface ipv4 show interfaces**命令，可以查看计算机上网络接口的摘要信息，参考如下实例：

Idx	Met	MTU	State	Name
1	50	4294967295	connected	Loopback Pseudo-Interface 1
7	10	1500	connected	Local Area Connection

上面的实例中，计算机有两个网络接口，Loopback Pseudo-Interface已连接。接口索引为1，接口metric为50，以太网最大传输单元（MTU）为4,294,967,295字节。本地区域连接接口业已连接，接口索引为7，接口metric为10，MTU为1,500字节。

使用Ethernet II封装时，MTU为1500表示在网络上传输的每一个数据块大小为1500字节，其中20字节用于IP头，其余的1480字节用于IP载荷。因而，65,535字节大小的IP数据报需要分割为很多小数据块以便在网络上传输，之后在目的节点上重组。

计算机网线两端都插好时，网络接口的状态显示为已连接，两端都没插好或者某一端没插好，网络接口的状态会显示为断开连接。

17.3.3 操作 TCP Internet 控制与错误消息

在IP上发送的每一个数据包都是数据报，也就是说，是一个尚未得到应答、尚未分配序列号的消息，该消息由路由器转发到目的IP地址。接受数据报的每个路由器会决定转发该数据报的最佳路径，这意味着，不同的数据报在发送端IP地址（源节点）与目的端IP地址（目的节点）之间传送时的路由可能是不一致的，也意味着不同数据报的应答消息所走的路由也是不一致的。

尽管IP为IP数据报提供端对端的传输能力，但不会提供报告路由信息或传输错误的机制。错误与控制消息是通过Internet控制消息协议（ICMP）进行追踪与处理的，通过在命令提示符中键入**netsh interface ipv4 show icmp**，可以查看ICMP的统计资料，该命令的输出类似于如下格式：

```

MIB-II ICMP Statistics
-----

```

```

INPUT
Messages:    20302
Errors:      120
Destination Unreachable:    45
Time Exceeded:    88
Parameter Problems:    0
Source Quench:    4
Redirects:    6
Echo Requests :    966
Echo Replies :    966
Time Stamp Requests:    0
Time Stamp Replies:    0
Address Mask Requests:    0
Address Mask Replies:    0

```

```

OUTPUT
Messages:    20302
Errors:      120
Destination Unreachable:    45
Time Exceeded:    88
Parameter Problems:    0
Source Quench:    4
Redirects:    6
Echo Requests:    966
Echo Replies:    966
Time Stamp Requests:    0
Time Stamp Replies:    0
Address Mask Requests:    0
Address Mask Replies:    0

```

提示 对提供摘要统计信息的Interface IPv4与IPv6 SHOW命令，可以将参数Rr=设置为刷新周期的秒数。比如，如果需要网络接口统计资料每隔30秒自动刷新，可以键入命令**netsh interface ipv4 show icmp rr=30**。设置刷新速率后，可以按Ctrl+C退出该命令，以免产生更多的更新行为。

前面的实例中，可以查看接收的IP数据报消息（输入消息）与发送的IP数据报消息（输出消息）的详细统计资料。如果确定自己需要寻找哪些信息，对这些统计信息进行解码是容易的。最基本的IP数据报消息是*Echo*消息，*Echo*消息用于向IP节点发送简单的消息，并等待接收方向发送方发回一个*Echo Reply*消息。很多TCP/IP网络命令使用*Echo*消息与*Echo Reply*消息来提供关于可达性与到目标IP节点的路径等信息。

在IP数据报传输过程中（出站与入站）发生的任何错误都被记录为错误。IP会尝试对IP数据报进行best-effort传输，以便将其传送到目的节点。如果在传输路径或目的节点处发生路由或传输错误，则路由器或目的节点会丢弃存在问题的数据报，并发送一条“目的地址不可达”的ICMP消息来报告这一错误。

在IP数据报发送之前，会设置一个生存期值（TTL），该值代表的是该数据报在源端与目的端之间可以经过的最大跳数。数据报的TTL值过期时，会向该数据报的发送端返回一条“Time exceeded”消息。典型情况下，这意味着在源端点与目的端点之间存在多于预期的链路，也可能表明网络中出现了路由环路的情况——在路由器包含了错误的路由信息，以至于无法将数据报发送到其目的端点时，就

导致路由环路的出现。

如果在处理IP数据报的IP头时出错，路由器或目的节点会发送“Parameter Problem”消息。IP头错误会导致该IP数据报被丢弃，如果没有其他ICMP消息可用于描述已发生的错误，则会将“Parameter Problem”消息发回到源节点。典型情况下，这一消息表明IP头的格式不正确或IP选项字段中存在错误的参数。

路由器变得阻塞时，不管是因为流量的突增，还是链路传输速度慢或者流量散发，抑或是资源不充足，都会导致丢弃入站的IP数据报，并且由路由器向IP数据报的源端发送一条“Source Quench”消息，声称数据报发送过快以至于无法及时处理，目的节点也可能出于同样的原因向源端发送“Source Quench”消息。这种消息并不会为每一个被丢弃的数据报发送，而是为某些消息段，Internet工程任务组（IETF）RFC 1812的建议是不要对所有丢弃的数据报都发送这一消息，因为这会在已经拥挤的线路上制造更多的流量。如果源端收到了“Source Quench”消息，就会以较慢的传输速率发送相关的TCP段，以避免造成拥塞。

使用子网时，IP地址的第一部分并不能用于确定子网掩码。要发现其子网掩码，IP节点会向已知的路由器发送“地址请求”消息，或使用面向所有子网的广播地址（或受限的广播地址）。对该消息进行响应的路由器会发送一条“地址应答”消息，其中包含了收到“地址请求”消息的网段的子网掩码。如果某IP节点不知道自己的IP地址，也可以发送一条源地址为0.0.0.0的“地址请求”消息，此消息的接收者会假定源IP节点使用基于类的子网掩码并据此使用广播进行回应。

在数据经由TCP传输之前，接收方会通告自身每次最多可以接受多少数据，该值称为TCP窗口大小。进行数据传输时，TCP窗口大小决定了发送方等待来自接收方的应答之前可以传输的数据总量。TCP窗口大小是一个16位的字段，因此其最大的接收窗口大小为65535字节。也就是说，源节点在单一的TCP窗口中至多可以发送这样大小的数据。通过使用TCP window scale选项，接收者可以使用更大的窗口，最大大约可至1GB。

要计算重传超时值(RTO)，TCP需要追踪TCP段之间的TCP往返传输时间（以正在进行的为基础）。通常，为发送数据的每个完整窗口，都需要计算RTO。在很多网络环境下，这种方法都起到较好的效果，并可以防止数据的重发。然而，在高带宽的网络环境或延迟较大的网络环境下，这一技术并不能达到良好效果。为每个窗口进行的数据采样不能正确确定当前RTO，也不能防止不必要的数据重发。

要计算任意TCP段的RTT与RTO，可以在基于本地时钟的时间戳请求消息中包含时间戳值。TCP段中数据的应答消息会回显时间戳，使用回显时间戳与TCP段应答消息到达时间可以计算RTT。这些消息记录为时间戳请求与时间戳应答。

最后一种ICMP消息类型是在故障排除时使用的重定向消息，重定向消息用于通知IP数据报的发送者存在哪条到目的节点的更优化路由路径。由于大多数主机维护最小的路由表，这一信息可用于提高消息路由的效率，并减少传输的时间与错误。因此，发现重定向消息时，可以知道网络流量被重定向到目的节点。

17.3.4 检查分片、重组、错误消息的详细信息

要对IP数据报的分片与重组有更深入的了解，可以键入命令**netsh interface ipv4 show ipstats**，其输出类似于如下格式：

```
MIB-II IP Statistics
```

```
-----
```

```

Forwarding is:    Enabled
Default TTL:     128
In Receives:    24219
In Header Errors: 0
In Address Errors: 250
Datagrams Forwarded: 0
In Unknown Protocol: 0
In Discarded:    0
In Delivered:    23969
Out Requests:    20738
Routing Discards: 0
Out Discards:    0
Out No Routes:   0
Reassembly Timeouts: 60
Reassembly Required: 0
Reassembled Ok:  0
Reassembly Failures: 0
Fragments Ok:    0
Fragments Failed: 0
Fragments Created: 0

```

提示 你可以自动刷新这些统计信息，这需要为上面的命令添加 **Rr=RefreshRate**。其中，*RefreshRate* 为刷新时间间隔（以秒为计数单位）。

可以看出，IP统计信息中包含了默认的TTL值（用于本计算机上创建的待传输的出站数据包）。上面的实例中，TTL值为128，也就是说，本计算机到目的计算机之间至多可有128个链路。如果数据包使用了超过该值的跳数，则该数据包会被丢弃，并向源计算机发送一条“Time Exceeded”消息。

In Receives值指明已接收多少入站数据包，实际使用的数据包数由In Delivered值代表，两个值之间的差异是入站数据包，包括下面列举的。

- ☐ 接收但含错，在In Header Errors或In Address Errors中指定。
- ☐ 转发到其他IP节点，在Datagrams Forwarded中指定。
- ☐ 使用未知协议，在In Unknown Protocol中指定。
- ☐ 丢弃，比如数据包的TTL过期，在In Discarded中指定。

上面的实例中，在In Receives与In Delivered之间有250个数据包的差异，因为这250个入站的数据包带有地址错误。

上面的输出中也记录了出站数据包的数量与处置。出站的数据包数量被列为Out Requests，传输时如果出错，则会根据类型进行记录。如果路由器或其他节点发回“目的不可达”消息，通常会记录为“Routing Discard”。其他类型的错误消息，比如“Parameter Problem”消息或“Source Quench”消息，可能被记录为Routing Discards或Out Discards。如果没有出站的路由或返回“No Route”消息，则数据包可能被记录为Out No Routes。

数据通过路由器传输到本地网络之外时，典型情况下会被分片传输，并在目的端重组。上面的实例中，原始数据包的重组统计信息与接收分片的状态信息也包含在输出中。

17.3.5 检查当前的 TCP 与 UDP 连接

防火墙与代理服务器都可能会影响连接到本地或远程系统的能力。典型情况下，管理员需要打开

一些TCP、UDP端口，以便本地网络或远程网络上的其他计算机与本计算机进行远程通信。每一类型的应用程序或工具都需要打开不同的端口，已知服务使用的TCP、UDP端口的完整列表存储在`%SystemRoot%\System32\Drives\Etc\Services`。

然而有时候，待操作的工具并没有已知的服务与其关联，需要对其进行实验才能确定使用的是哪个TCP或UDP端口。一种有效的方法是启动该工具，之后使用TCP、UDP listener查看哪些端口状态变为活跃。

1. 操作TCP

TCP端口采用被动打开方式，其基本的含义是说TCP端口可以被主动接受请求。客户端需要使用TCP端口时，必须尝试与该端口建立TCP连接。TCP连接是IP网络上两个端点之间的双向连接，使用应用层协议，TCP端点是由IP地址与TCP端口进行标识的。TCP连接包括一个本地TCP端点与一个远程TCP端点，可用于识别本地计算机到本地计算机的回环连接，也可以用于识别本地计算机到网络上其他远程计算机之间的标准TCP连接。TCP连接是通过3次握手建立的，包括下面3个主要步骤。

- (1) 需要使用TCP端口的客户端发送一个活跃开放请求（SYN）。
- (2) 接收端对SYN请求进行应答，发回一个SYN-ACK消息。
- (3) 客户端向TCP端口发送一个最终的应答请求（ACK）完成3次握手。

经由TCP发送的数据通常会被分段，这些分段数据是以IP数据报的形式（包括TCP头与TCP数据）发送的。建立连接后，最大分段大小（MSS）也会被相应地设置。典型情况下，MSS最大值为65,495字节，这是IP数据报大小（65,535字节）与最小IP头（20字节）、最小TCP头（20字节）之间进行减操作得到的。技术上讲，SYN、SYN-ACK、ACK消息实际上是SYN、SYN-ACK、ACK段。

通过**netsh interface ipv4 show tcpconn**命令，可以查看当前TCP连接的统计信息。要自动刷新这些统计信息，可以使用参数**Rr=RefreshRate**。输出信息将展示哪些TCP端口处于监听状态，哪些端口已建立连接，哪些端口处于等待状态，如下面所示：

MIB-II TCP Connection Entry

Local Address	Local Port	Remote Address	Remote Port	State
0.0.0.0	42	0.0.0.0	18520	Listen
0.0.0.0	53	0.0.0.0	16499	Listen
0.0.0.0	88	0.0.0.0	45165	Listen
0.0.0.0	135	0.0.0.0	2176	Listen
0.0.0.0	389	0.0.0.0	2256	Listen
0.0.0.0	1025	0.0.0.0	43054	Listen
0.0.0.0	1026	0.0.0.0	35016	Listen
0.0.0.0	1028	0.0.0.0	53398	Listen
0.0.0.0	3069	0.0.0.0	43189	Listen
0.0.0.0	3268	0.0.0.0	43230	Listen
0.0.0.0	3269	0.0.0.0	36957	Listen
127.0.0.1	389	127.0.0.1	1033	Established
127.0.0.1	389	127.0.0.1	1034	Established
127.0.0.1	389	127.0.0.1	1035	Established
127.0.0.1	389	127.0.0.1	1039	Established
127.0.0.1	1033	127.0.0.1	389	Established
127.0.0.1	1034	127.0.0.1	389	Established
127.0.0.1	1035	127.0.0.1	389	Established
127.0.0.1	1039	127.0.0.1	389	Established

127.0.0.1	3073	0.0.0.0	10251	Listen
192.168.1.50	135	192.168.1.56	1040	Listen
192.168.1.50	139	0.0.0.0	12369	Listen
192.168.1.50	389	192.168.1.50	3287	Established
192.168.1.50	3287	192.168.1.50	389	Established
192.168.1.50	3289	192.168.1.50	135	Wait
192.168.1.50	290	192.168.1.50	1025	Wait

条目0.0.0.0代表的是TCP广播地址，条目127.0.0.1代表的是本地计算机使用的本地回环端口，你也可以看计算机使用的物理地址上回连到该计算机的条目。这里，这些地址是指设置为192.168.1.50的本地与远程地址。最应该引起关注的是那些远程IP地址与本地IP地址不同的条目，这些条目表示的是本地计算机到其他系统与网络的连接。

Local Port列与Remote Port列展示了本地TCP端口如何映射到远程TCP端口。比如，上面的输出信息中，IP地址192.168.1.50的本地端口135映射为远程计算机（IP地址为192.168.1.56）的1040端口。每个TCP连接都有一个状态值，表17-3总结了最常见的状态值。

表17-3 TCP连接状态

状 态	描 述
Closed	当前尚无TCP连接
Listen	某应用程序层协议发起了一个被动的开放函数调用，允许接受对指定端口的入站连接请求，此时尚未产生TCP流量
Syn Sent	使用应用程序层协议的客户端发起了主动的开放函数调用（SYN），创建并发送了TCP连接3次握手的第一部分数据
Syn Rcvd	使用应用程序层协议的客户端接收到了SYN请求，并发送一条应答消息（SYN-ACK）
Established	3次握手中的最后一个ACK已经被接收，TCP连接已经建立，此时可以双向传递数据
Wait	TCP连接已被终止，并由本地客户端与远程客户端进行了应答（FIN-ACK）

通过**netsh interface ipv4 show tcpstats**命令，可以查看附加的TCP统计信息，其输出应该类似于如下格式：

MIB-II TCP Statistics

Timeout Algorithm:	Van Jacobson's Algorithm
Minimum Timeout:	10
Maximum Timeout:	4294967295
Maximum Connections:	Dynamic
Active Opens:	182
Passive Opens:	174
Attempts Failed:	6
Established Resets:	226
Currently Established:	46
In Segments:	410814
Out Segments:	410448
Retransmitted Segments:	2811
In Errors:	0
Out Resets:	171

TCP统计信息包括如下一些详细资料。

- 超时的最大值与最小值。

- ❑ 计算机上启动TCP/IP网络以来主动与被动开放总数。
- ❑ 尝试建立但失败的连接。
- ❑ 建立之后重置的连接。
- ❑ 当前建立的连接总数。
- ❑ 发送的TCP段总数（入段）与接受的TCP段总数（出段）。
- ❑ 必须重发的TCP段数。
- ❑ 已经接收的包含错误（入错误）的TCP段总数。

2. 操作UDP

与面向连接的TCP协议不同的是，UDP是一种无连接的协议，也就是说，UDP消息发送之前并不需要预先建立连接。UDP端口与TCP端口是完全不同的，即便端口号是一样的。由于UDP消息没有序列号与应答消息，因此是不可靠的，而与之相反，TCP连接是非常可靠的。操作UDP协议时，只有一个本地地址与本地端口组合，代表的是被监听的端口。通过**netsh interface ipv4 show udpconn**命令，可以查看相关的监听者条目，其输出信息类似于如下格式：

```
MIB-II UDP Listener Entry
Local Address      LocalPort
-----
0.0.0.0           42
0.0.0.0           445
0.0.0.0           500
0.0.0.0           1030
0.0.0.0           1032
0.0.0.0           1701
0.0.0.0           3002
0.0.0.0           3103
0.0.0.0           3114
0.0.0.0           4500
127.0.0.1         53
127.0.0.1         123
127.0.0.1         1036
127.0.0.1         3101
127.0.0.1         3102
192.168.1.50      53
192.168.1.50      67
192.168.1.50      137
192.168.1.50      138
192.168.1.50      389
192.168.1.50      464
192.168.1.50      2535
```

条目0.0.0.0代表的是UDP广播地址，条目127.0.0.1代表的是本地计算机使用的本地回环端口，网络适配器物理IP地址条目为用于监听连接的端口。

UDP消息是以IP数据报的形式发送的，其中包含了UDP头与UDP消息数据。通过**netsh interface ipv4 show udpstats**命令，可以查看附加的UDP统计信息，其输出应该类似于如下格式：

```
MIB-II UDP Statistics
-----
In Datagrams:      42640
```

```
In Invalid Port:      732
In Erroneous Datagrams: 20
Out Datagrams:      72217
```

UDP统计信息包括如下一些详细资料。

- ☐ UDP端口接收的数据报总数。
- ☐ 无效端口接收并丢弃的数据报总数。
- ☐ 接收并丢弃的含错数据报总数。
- ☐ 通过UDP端口发送的数据报总数。

17.4 排除 TCP/IP 网络故障

TCP/IP网络的问题可能是难以追寻的，这也是为什么会有那么多工具来帮助确定问题的所在。在进行故障排除之前，要确保对17.3节中的概念与过程有清晰的理解和掌握，其中所讨论的工具与技术有助于发现与诊断一些最复杂的TCP/IP网络问题。除该节中的讨论之外，你也可以用本节中讨论的工具与技术来对网络联通性与配置问题进行故障排除。

17.4.1 查看诊断信息

很多TCP/IP网络问题都与网络组件的不正确配置相关，要获取计算机网络配置与负载快照，最快、最简单的方法是使用Netsh Ras Diagnostics上下文中的命令。尽管这些命令的设计目标是对远程访问相关问题进行故障排除，但实际上也可以用于对一般性的网络问题进行故障排除。

Netsh Ras Diagnostics上下文提供了不同的Show命令（用于操作指定类型的诊断信息）与Set命令（用于配置诊断记录、追踪、报告）。然而，一般不会只关注某一个很狭窄的领域进行诊断，而是希望生成网络整体情况的诊断报告。为此，可以键入命令**netsh ras diagnostics show all type=file destination=FileName verbose=enabled**，其中，*FileName*为该命令将要生成的HTML文件名。

在IE中查看生成的诊断文件时，会发现几个日志的内容，包括远程访问踪迹日志、调制解调器踪迹日志、连接管理器日志、IPsec踪迹日志以及远程访问事件日志等。还可以看到对远程访问与网络必需组件是否安装的检查、已安装网络组件的详细列表、远程访问所用到的注册表键的当前值等信息。尽管存在这么多信息，但最应该集中关注的是如下一些命令行工具的输出信息。

- ☐ **arp.exe -a**，显示了地址解析协议的相关条目。
- ☐ **ipconfig.exe /all**，显示了所有网络接口的配置信息。
- ☐ **ipconfig.exe /displaydns**，显示了DNS解析器缓存中的内容。
- ☐ **route.exe print**，显示IPv4与IPv6路由表的内容。
- ☐ **net.exe start**，列出所有运行中的Windows服务。
- ☐ **netstat.exe -e**，列出用于数据包传输的网络接口统计信息。
- ☐ **netstat.exe -o**，根据协议列出活跃连接，包括TCP连接与UDP连接。
- ☐ **netstat.exe -s**，列出IPv4、IPv6、ICMPv4、ICMPv6、TCP for IPv4、TCP for IPv6、UDP for IPv4、UDP for IPv6的摘要统计信息。
- ☐ **netstat.exe -n**，根据地址与端口列出活跃连接，包括TCP连接与UDP连接。
- ☐ **nbtstat.exe -c**，列出运行于TCP/IP之上的NetBIOS缓存内容。
- ☐ **nbtstat.exe -n**，列出本地NetBIOS名。

- ❑ nbtstat.exe -r, 列出了由广播与WINS解析的名。
- ❑ nbtstat.exe -S, 列出了NetBIOS会话表及其目的IP地址。
- ❑ netsh.exe dump, 对网络配置信息进行完全的转储。转储文件可用于查看网络配置, 或以后用于重建网络配置。

注解 这些命令可以在命令提示符中直接运行, 但要记住的是, 通过Netsh, 可以对网络中存在的问题进行远程故障排除, 而不是必须物理接触该计算机或者通过远程桌面远程登录该计算机——只需要启动Netsh并使用-R参数指定远程计算机名, 就可以着手诊断存在的问题了。

17.4.2 诊断常规的计算机配置问题

作为故障诊断排除工作的一部分, 你可能需要查看计算机与操作系统的详细配置信息。对此, 最佳的一个途径就是访问WMI对象 (用于追踪相关的配置设置信息)。比如, Win32_ComputerSystem对象用于追踪计算机整体配置信息, Win32_OperatingSystem对象用于追踪操作系统整体配置信息。

Windows PowerShell是可用于操作WMI的一种途径。在Windows PowerShell中, 可以使用Get-WmiObject来获取待操作的WMI对象。通过将对象重定向到Format-List*, 可以列出对象的所有属性与值。操作WMI对象时, 可能需要操作根命名空间, 这可以通过将-Namespace参数设置为root/cimv2实现。通过-Computer参数, 可以指定待操作的计算机。如果待操作的是本地计算机, 则可以使用句点(.)而非计算机名来指定。

在Windows PowerShell提示符中, 输入如下命令, 可以检查Win32_ComputerSystem对象及其属性, 来获取计算机配置相关的摘要信息:

```
Get-WmiObject -Class Win32_ComputerSystem -Namespace root/cimv2
-ComputerName . | Format-List *
```

如果想将输出信息保存到某个文件, 只需将输出信息重定向到该文件即可。下面的实例中, 将输出信息重定向到工作目录下名为computer_save.txt的文件中:

```
Get-WmiObject -Class Win32_ComputerSystem -Namespace root/cimv2
-ComputerName . | Format-List * > computer_save.txt
```

命令清单17-2列出了由上面命令获取的详细的计算机信息。

命令清单17-2 计算机配置信息

```
AdminPasswordStatus      : 1
BootupState               : Normal boot
ChassisBootupState       : 3
KeyboardPasswordStatus   : 2
PowerOnPasswordStatus    : 1
PowerSupplyState         : 3
PowerState                : 0
FrontPanelResetStatus    : 2
ThermalState              : 3
Status                   : OK
Name                     : MAILSERVER25
PowerManagementCapabilities :
```



```

PowerManagementSupported      :
__GENUS                        : 2
__CLASS                        : Win32_ComputerSystem
__SUPERCLASS                   : CIM_UnitaryComputerSystem
__DYNASTY                       : CIM_ManagedSystemElement
__RELPATH                      : Win32_ComputerSystem.Name="MAILSERVER25"
__PROPERTY_COUNT               : 58
__DERIVATION                   : {CIM_UnitaryComputerSystem,
CIM_ComputerSystem, CIM_System, CIM_LogicalElement...}
__SERVER                       : MAILSERVER25
__NAMESPACE                    : root\cimv2
__PATH                         : \\MAILSERVER25\root\
cimv2:Win32_ComputerSystem.Name="MAILSERVER25"
AutomaticManagedPagefile      : True
AutomaticResetBootOption      : True
AutomaticResetCapability       : True
BootOptionOnLimit              :
BootOptionOnWatchDog           :
BootROMSupported               : True
Caption                        : MAILSERVER25
CreationClassName              : Win32_ComputerSystem
CurrentTimeZone                 : -420
DaylightInEffect               : True
Description                     : AT/AT COMPATIBLE
DNSHostName                    : MAILSERVER25
Domain                         : cpandl.com
DomainRole                     : 5
EnableDaylightSavingsTime      : True
InfraredSupported              : False
InitialLoadInfo                :
InstallDate                    :
LastLoadInfo                   :
Manufacturer                   : Dell Inc.
Model                         : Dimension XPS
NameFormat                     :
NetworkServerModeEnabled       : True
NumberOfLogicalProcessors      : 2
NumberOfProcessors             : 1
OEMLogoBitmap                  :
OEMStringArray                 : {www.dell.com}
PartOfDomain                   : True
PauseAfterReset                : -1
PCSystemType                   : 5
PrimaryOwnerContact            :
PrimaryOwnerName               : Windows User
ResetCapability                 : 1
ResetCount                     : -1
ResetLimit                     : -1
Roles                          : {LM_Workstation, LM_Server,
Primary_Domain_Controller, Timesource...}
SupportContactDescription      :
SystemStartupDelay             :
SystemStartupOptions           :

```

```

SystemStartupSetting      :
SystemType                : x64-based PC
TotalPhysicalMemory       : 3755343872
UserName                  : CPANDL\williams
WakeUpType                : 6
Workgroup                 :
Scope                    : System.Management.ManagementScope
Path                     : \\Server52\root\cimv2:
Win32_ComputerSystem.Name="Server52"
Options                   : System.Management.ObjectGetOptions
ClassPath                 : \\Server52\root\cimv2:Win32_ComputerSystem
Properties                 : {AdminPasswordStatus...}
SystemProperties          : {__GENUS, __CLASS, __SUPERCLASS...}
Qualifiers                : {dynamic, Locale, provider, UUID}
Site                     :
Container                 :

```

表17-4对输出信息中相应配置条目及其含义进行了总结。

表17-4 计算机配置条目及其含义

属 性	描 述
AdminPasswordStatus	管理员口令状态。值1代表禁用，值2代表激活，值3代表未实现，值4代表未知
AutomaticManagedPagefile	指明该计算机的页面文件是否由操作系统来管理
AutomaticResetBootOption	指明自动重设引导选项是否激活
AutomaticResetCapability	指明自动重设是否激活
BootOptionOnLimit	达到ResetLimit值时采取的系统动作。1代表保留，2代表操作系统，3代表系统工具，4代表不重新引导
BootOptionOnWatchDog	在观测的计时器时间已到时采取的重新引导操作。1代表保留，2代表操作系统，3代表系统工具，4代表不重新引导
BootROMSupported	指明是否支持引导ROM
BootupState	指明系统启动方式。值包括正常引导、安全模式引导、带网络连接的安全模式引导
Caption	系统名
ChassisBootupState	系统机箱的启动状态。值为1代表其他，2代表未知，3代表安全，4代表警告，5代表关键，6代表不可恢复
ClassPath	WMI对象类路径
Container	与对象相关联的容器
CreationClassName	导出对象的类名
CurrentTimeZone	计算机从协调世界时偏离的分钟数
DaylightInEffect	指明是否激活夏时制模式
Description	计算机的描述信息
DNSHostName	根据DNS获取的服务器名
Domain	计算机所属域的域名
DomainRole	计算机的域角色。0代表独立工作站，1代表成员工作站，2代表独立服务器，3代表成员服务器，4代表备份域控制器，5代表主域控制器
EnableDaylightSavingsTime	指明是否激活了夏令时时间。如果取值为TRUE，则系统会改变为DST启动或终止后的前一个小时或后一个小时。如果取值为FALSE，则不做这种改变

(续)

属 性	描 述
FrontPanelResetStatus	计算机上重置按钮的硬件安全设置。0代表禁用, 1代表激活, 2代表未实现, 3代表未知
InfraredSupported	指明计算机上是否存在红外 (IR) 端口
InitialLoadInfo	发现初始负载设备或引导服务 (用来请求操作系统启动) 必需的数据
InstallDate	计算机的安装时间
KeyboardPasswordStatus	指明键盘口令状态。0代表禁用, 1代表激活, 2代表未实现, 3代表未知
LastLoadInfo	InitialLoadInfo属性的数组条目, 其中保存了引导当前加载的操作系统的相应数据
Manufacturer	计算机制造商名
Model	制造商给定的产品名
Name	计算机名
NameFormat	识别计算机系统名是如何生成的
NetworkServerModeEnabled	指明是否激活了网络服务器模式
Number Of Logical Processors	处理器核的数量。如果计算机有两个处理器, 每个处理器有4个核, 则逻辑处理器数量为8。如果计算机是超线程体系结构, 则逻辑处理器的数量也可能超过实际处理器的数量
NumberOfProcessors	计算机上激活的处理器数
OEMLogoBitmap	标识OEM的logo的位图
OEMStringArray	列出OEM设置的描述性的字符串
PartOfDomain	指明计算机是否为某个域的一部分。如果值为TRUE, 则该计算机属于某个域。如果值为FALSE, 则计算机属于某个工作组
Options	列出了管理对象选项
Path	标识对象类的全WMI路径
PauseAfterReset	在系统关闭电源再启动或重设后到重新引导启动前的时间延迟 (以毫秒为计数单位), 值为-1代表没有时间延迟
PCSystemType	指明计算机的类型。0代表未指定, 1代表桌面系统, 2代表移动计算机, 3代表工作站, 4代表企业级服务器, 5代表SOHO服务器, 6代表应用设备PC, 7代表性能服务器, 8代表Role Maximum
PowerManagementCapabilities	逻辑驱动器的电源管理能力。0代表未知, 1代表不支持, 2代表禁用, 3代表激活, 4代表自动进入省电模式, 5代表电源状态可设置, 6代表支持电源循环装置, 7代表支持时控开机
PowerManagementSupported	指明设备的电源是否可以管理
PowerOnPasswordStatus	开机口令状态。0代表禁用, 1代表激活, 2代表未实现, 3代表未知
PowerState	指明计算机的当前电源状态。0代表未知, 1代表电量充满, 2代表节电-低电源模式, 3代表节电-待机, 4代表节电-未知, 5代表关机再开机, 6代表关机, 7代表节电-警告
PowerSupplyState	上一次引导时电源箱状态。1代表其他, 2代表未知, 3代表安全, 4代表告警, 5代表关键, 6代表不可恢复
PrimaryOwnerContact	计算机所有者的联系信息
PrimaryOwnerName	系统所有者名
Properties	列出对象的所有属性
Qualifiers	列出对象的任意qualifiers

(续)

属 性	描 述
ResetCapability	指明计算机是否可以用电源或重置按钮（或其他硬件方式）重置。1代表其他，2代表未知，3代表禁用，4代表激活，5代表不可恢复
ResetCount	自上次人为重置后自动重置的次数，值为-1表示未知
ResetLimit	系统尝试重置的连续次数，值-1代表未知
Roles	系统角色
Scope	列出管理对象范围
Site	与该对象相关联的站点
Status	计算机当前状态，取值为良好、错误、降级、未知、Pred Fail、启动、终止、服务
SupportContactDescription	列出计算机技术支持的联系信息
SystemProperties	列出系统属性
SystemStartupDelay	启动延迟（以秒为计数单位）
SystemStartupOptions	列出计算机的启动选项
SystemStartupSetting	默认启动配置文件的索引
SystemType	计算机体系结构类型，比如基于X86的PC或64位的Intel PC
ThermalState	上次引导时系统机箱的热状态。1代表其他，2代表未知，3代表安全，4代表告警，5代表关键，6代表不可恢复
TotalPhysicalMemory	物理内存的字节总数
UserName	当前登录用户名
WakeUpType	导致系统开机的事件。值为0代表保留，1代表其他，2代表未知，3代表APM定时器，4代表Modem Ring，5代表LAN远程，6代表电源开关，7代表PCI PME#，8代表AC Power Restored
Workgroup	如果计算机属于某个工作组，则列出工作组名

可以看出，通过这些详细的配置信息，可以获知计算机的配置情况。对操作系统同样如此，在Windows PowerShell提示符中，输入如下命令，可以获取操作系统的详细资料：

```
Get-WmiObject -Class Win32_OperatingSystem -Namespace root/cimv2
-ComputerName . | Format-List *
```

命令清单17-3列出了上面命令的输出信息实例。与此前讨论的一样，你可以将这些输出信息重定向保存到相应文件中。

命令清单17-3 详细的操作系统配置输出

```
Status : OK
Name : Microsoft® Windows Server
      © 2008 Enterprise |C:\Windows|\Device\Harddisk1\Partition1
FreePhysicalMemory : 679172
FreeSpaceInPagingFiles : 3749368
FreeVirtualMemory : 2748020
__GENUS : 2
__CLASS : Win32_OperatingSystem
__SUPERCLASS : CIM_OperatingSystem
__DYNASTY : CIM_ManagedSystemElement
```

```

__RELPATH : Win32_OperatingSystem=@
__PROPERTY_COUNT : 65
__DERIVATION : {CIM_OperatingSystem, CIM
_LogicalElement, CIM_ManagedSystemElement}
__SERVER : MAILSERVER25
__NAMESPACE : root\cimv2
__PATH : \\MAILSERVER25\root\cimv2
:Win32_OperatingSystem=@
BootDevice : \Device\HarddiskVolume1
BuildNumber : 6001
BuildType : Multiprocessor Free
Caption : Microsoft® Windows
Server® 2008 Enterprise
CodeSet : 1252
CountryCode : 1
CreationClassName : Win32_OperatingSystem
CSCreationClassName : Win32_ComputerSystem
CSDVersion : Service Pack 1, v.745
CSName : MAILSERVER25
CurrentTimeZone : -420
DataExecutionPrevention_32BitApplications : True
DataExecutionPrevention_Available : True
DataExecutionPrevention_Drivers : True
DataExecutionPrevention_SupportPolicy : 3
Debug : False
Description :
Distributed : False
EncryptionLevel : 256
ForegroundApplicationBoost : 2
InstallDate : 20080917143704.000000-480
LargeSystemCache :
LastBootUpTime : 20080804124518.375199-420
LocalDateTime : 20080804183034.619000-420
Locale : 0409
Manufacturer : Microsoft Corporation
MaxNumberOfProcesses : 4294967295
MaxProcessMemorySize : 8589934464
UILanguages : {en-US}
NumberOfLicensedUsers :
NumberOfProcesses : 95
NumberOfUsers : 3
OperatingSystemSKU : 10
Organization :
OSArchitecture : 64-bit
OSLanguage : 1033
OSProductSuite : 274
OSType : 18
OtherTypeDescription :
PAEEnabled :
PlusProductID :
PlusVersionNumber :
Primary : True
ProductType : 2
QuantumLength : 1
QuantumType : 1

```



```

RegisteredUser      : Windows User
SerialNumber         :
ServicePackMajorVersion : 1
ServicePackMinorVersion : 0
SizeStoredInPagingFiles : 3974528
SuiteMask            : 274
SystemDevice         : \Device\HarddiskVolume2
SystemDirectory      : C:\Windows\system32
SystemDrive          : C:
TotalSwapSpaceSize   :
TotalVirtualMemorySize : 7591744
TotalVisibleMemorySize : 3667328
Version              : 6.0.6001
WindowsDirectory     : C:\Windows
Scope                : System.Management.ManagementScope
Path                 : \\Server52\root\cimv2:
Win32_OperatingSystem=@
Options              : System.Management.ObjectGetOptions
ClassPath             : \\CorpServer52\root\cimv2:
Win32_OperatingSystem
Properties            : {BootDevice...}
SystemProperties      : {__GENUS, __CLASS, __SUPERCLASS...}
Qualifiers            : {dynamic, locale, provider, singleton...}
Site                  :
Container             :

```

表17-5对输出信息中相应配置条目及其含义进行了总结。

表17-5 操作系统配置条目及其含义

属 性	描 述
BootDevice	Win32操作系统所在的磁盘驱动器
BuildNumber	操作系统的构建号
BuildType	操作系统的构建类型, 比如零售版或调试版或multiprocessor free版
Caption	操作系统名
ClassPath	WMI对象类路径
CodeSet	操作系统使用的代码页值
Container	与对象相关联的容器
CountryCode	操作系统使用的国家代码
CreationClassName	导出该对象的类名
CSCreationClassName	导出该计算机系统对象的类名
CSDVersion	指明计算机上是否安装了最新的服务补丁, 值为“空”表明尚未安装服务补丁
CSName	与该对象类相关联的计算机系统名
CurrentTimeZone	操作系统偏离格林尼治标准时间的分钟数, 可以为正数、负数或0
DataExtraction-Prevention_32BitApplications	指明是否为32位应用程序激活数据执行保护 (DEP)
DataExtraction-Prevention_Available	指明系统硬件是否支持数据执行保护 (DEP)
DataExtraction-Prevention_Drivers	指明设备驱动程序是否激活数据执行保护 (DEP)

(续)

属 性	描 述
DataExtraction-Prevention_SupportPolicy	指明使用的DEP支持策略。0代表不支持, 2代表仅支持基本的Windows程序与服务, 3代表支持所有程序(特殊指明的除外)
Debug	指明操作系统是否为调试版。如果为TRUE, 则安装的是调试版的 User.exe
Description	Windows操作系统的描述信息
Distributed	指明操作系统是否分布在多个计算机系统节点上。如果是, 则这些节点应该组成一个簇
EncryptionLevel	安全交易的加密级别, 40比特、128比特或n比特
Foreground-ApplicationBoost	设置前台应用程序的优先级。应用程序的启动是通过为其分配更多处理器时间实现的。值为0代表没有分配处理器时间, 1代表最少, 2代表最多(默认情况)
FreePhysicalMemory	未使用但可用的物理内存总量(以KB为计数单位)
FreeSpaceInPagingFiles	操作系统页面文件空闲空间总量, 空闲空间用完时就会交换页面
FreeVirtualMemory	未使用但可用的虚拟内存总量(以KB为计数单位)
InstallDate	操作系统安装时间
LargeSystemCache	指明程序或系统缓存的内存使用是否进行了优化。值为0表示程序的内存使用进行了优化, 值为1代表系统缓存的内存使用进行了优化
LastBootUpTime	操作系统上次引导时间
LocalDateTime	计算机的本地日期与时间
Locale	操作系统使用的语言识别符
Manufacturer	操作系统制造商。对Win32系统, 该值为Microsoft Corporation
MaxNumber-OfProcesses	操作系统可以支持的进程上下文最大编号。如果没有固定的最大值, 则该值为0
MaxProcess-MemorySize	可分配给进程的最大内存容量(以KB计数), 值为0代表没有上限
MUILanguages	支持的用户界面语言
Name	操作系统实例名
NumberOfLicensedUsers	操作系统的用户许可证编号, 值为0代表无限制, 值为-1代表未知
NumberOfProcesses	系统上进程上下文的当前编号
NumberOfUsers	用户会话的当前编号
OperatingSystemSKU	操作系统产品类型指示器
Options	列出管理对象选项
Organization	操作系统注册用户的公司名集
OSArchitecture	操作系统体系结构, 32位或64位
OSLanguage	已安装的操作系统的语言版本
OSProductSuite	已安装的操作系统的产品套件
OSType	操作系统类型。1代表其他, 18代表Windows NT或后续版本
OtherTypeDescription	设置附加的描述信息, 在OSType = 1时使用
Path	对象类的全WMI路径
PAEEnabled	指明是否激活了物理地址扩展(PAE)
PlusProductID	Windows Plus! (如果已安装)的产品号
PlusVersionNumber	Windows Plus! (如果已安装)的版本号
Primary	指明是否为主操作系统

(续)

属 性	描 述
ProductType	操作系统产品类型。值为1代表工作站, 2代表域控制器, 3代表服务器
Properties	列出对象的所有属性
Qualifiers	列出对象的任意qualifier
QuantumLength	每处理器执行单元的时钟节拍数。1代表未知, 2代表1个时钟节拍, 3代表两个时钟节拍
QuantumType	处理器执行单元长度类型。1代表未知, 2代表固定的, 3代表可变。对可变长度, 前台应用程序与后台应用程序可以有不同的取值; 对固定长度, 前台应用程序与后台应用程序取值是一致的
RegisteredUser	操作系统注册用户姓名集
Scope	列出管理对象范围
SerialNumber	操作系统产品序列号
ServicePack-MajorVersion	计算机上安装的服务补丁的主版本号。如果尚未安装服务补丁, 则该值为0或NULL
ServicePack-MinorVersion	计算机上安装的服务补丁的从版本号。如果尚未安装服务补丁, 则该值为0或NULL
Site	对象相关联的站点
SizeStoredIn-PagingFiles	可以存储在操作系统页面文件的KB总量, 值为0表明不存在页面文件
Status	对象的当前状态, 包括良好、错误、未知、降级、Pred Fail、启动、终止、服务
SuiteMask	识别系统有哪些可用产品套件的位标志
SystemDevice	安装操作系统的物理磁盘分区
SystemDirectory	操作系统的系统目录
SystemDrive	安装操作系统的物理磁盘分区
SystemProperties	列出系统属性
TotalSwapSpaceSize	交换空间总量(以KB为计数单位)。如果交换空间与页面文件没有区分, 则该值可以未指定(NULL)
TotalVirtualMemorySize	虚拟内存总量(以KB为计数单位)
TotalVisibleMemorySize	操作系统可用物理内存总量(以KB为计数单位)
Version	操作系统版本号
WindowsDirectory	操作系统的Windows目录

诊断IP、DNS、WINS配置问题

Netsh Interface IPv4上下文提供了用于查看计算机上IP、DNS、WINS配置的命令, 下面给出了这些命令及其实例输出。

□ Netsh interface ipv4 show addresses。展示计算机上网络适配器使用的IP地址, 如下面实例所示:

```
Configuration for interface "Local Area Connection"
  DHCP enabled:                Yes
  IP Address:                   192.168.1.101
  Subnet Prefix:                 192.168.1.0/
24 (mask 255.255.255.0)
  Default Gateway:              192.168.1.1
  Gateway Metric:               0
```

```

InterfaceMetric:                    5

Configuration for interface "Local Area Connection 2"
  DHCP enabled:                     Yes
  IP Address:                       192.168.5.42
  Subnet Prefix:                    192.168.51.0/
24 (mask 255.255.255.0)
  Default Gateway:                  192.168.5.1
  Gateway Metric:                   0
  InterfaceMetric:                  5
Configuration for interface "Loopback Pseudo-Interface 1"
  DHCP enabled:                     No
  IP Address:                       127.0.0.1
  Subnet Prefix:                    127.0.0.0/8
(mask 255.0.0.0)
  InterfaceMetric:                  50

```

网络适配器是按序列出的。由于该计算机有两个网络适配器，因此除了回环网络接口外，还有两个条目。禁用的或不可用的网络适配器不会列出。

网关是根据网络适配器使用的顺序且以每个适配器为基础列出的。如果计算机有多个网络适配器，则每个已配置的网络适配器应该作为一个条目。对错误配置的网关（即不在同一个子网内），则没有符号。

- ❑ **Netsh interface ipv4 show dnsservers**。显示为计算机上网络适配器定义的DNS服务器，如下面实例所示：

```

Configuration for interface "Local Area Connection"
  DNS servers configured through DHCP: 192.168.1.120
                                         192.168.1.225
  Register with which suffix:          Primary only

Configuration for interface "Local Area Connection2"
  DNS servers configured through DHCP: 192.168.5.86
                                         192.168.5.124
  Register with which suffix:          Primary only

Configuration for interface "Loopback Pseudo-Interface 1"
  Statically Configured DNS Servers:   None
  Register with which suffix:          Primary only

```

每个已配置的DNS服务器是以其搜索顺序定义的，要确保使用了正确的IP地址，并且搜索顺序也是正确的。

- ❑ **Netsh interface ipv4 show winsservers**。显示为计算机上网络适配器定义的WINS服务器，如下面实例所示：

```

Configuration for interface "Local Area Connection"
  WINS servers configured through DHCP: 192.168.1.128
                                         192.168.1.144

Configuration for interface "Local Area Connection 2"
  WINS servers configured through DHCP: 192.168.5.45
                                         192.168.5.67

Configuration for interface "Loopback Pseudo-Interface 1"
  Statically Configured WINS Servers:  None

```

每个已配置的WINS服务器是以其搜索顺序定义的，要确保使用了正确的IP地址，并且搜索顺序也是正确的。

前面讨论的，你也可以在命令提示符中使用**Ipconfig/all**，用来显示所有TCP/IP配置信息。对DHCPv4，可以通过如下命令发布与更新IPv4配置：

```
ipconfig /release
ipconfig /renew
```

对DHCPv6，可以通过如下命令发布与更新IPv6配置：

```
ipconfig /release6
ipconfig /renew6
```

如果怀疑某计算机的DNS存在问题，则可以在命令提示符中键入命令**ipconfig/displaydns**，来显示DNS解析器缓存的内容。你可以在命令提示符中键入命令**ipconfig/flushdns**，来显示DNS解析器缓存的内容。要刷新所有DHCP租赁与再注册DNS名，可以在命令提示符中键入命令**ipconfig/registerdns**。

其他可用于对TCP/IP配置与连通性进行故障排除，包括下面列举的。

- **Tracert**。对计算机之间的连接路径进行追踪。
- **Ping**。确定是否可以建立网络连接。
- **Pathping**。上面两个命令的结合，对网络路由进行追踪，并提供丢包信息。

要检查连通性，可以使用上面的任意命令。在连通性出现问题时，输出信息将提供辅助确认信息。下面的实例中，计算机不能连接到指定的IP地址：

```
Pinging 192.168.1.1 with 32 bytes of data:
```

```
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

```
Ping statistics for 192.168.1.1:
```

```
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

上面的输出信息中，该计算机可能尚未连接到网络，或者网络配置是错误的。要记住的是，源节点与目的节点的Windows防火墙与其他防火墙都可能阻止上述网络行为。

要查看如何检查特定主机的网络连通性，参考如下实例。

追踪到192.168.1.100的路由信息：

```
tracert 192.168.1.100
```

尝试建立到Mailserver23的连接（根据主机名）：

```
ping mailserver23
```

追踪Mailserver23.cpandl.com的连接，并提供丢包信息：

```
pathping mailserver23.cpandl.com
```

要尝试确认到不同远程主机的连通性与TCP/IP配置，可以通过如下的某种方式实现。

- 对主机的IP地址、计算机名与完全限定域名进行Ping操作。根据IP地址、计算机名与完全限定

域名确定远程主机的连通性与名解析。

- 对主机本地回环地址进行Ping操作。确认网络适配器与TCP/IP是否进行了配置与激活。
- 对DHCP、DNS、WINS服务器进行Ping操作。确认网络适配器的DHCP、DNS、WINS服务器设置。
- 对网关进行Ping操作。确认网络适配器的默认网关设置。

计算机出现可能的连通性与配置问题时，应该立即对如下一些问题进行确认。

- 计算机是否连接到网络上？如果计算机可以连接到它的一个默认网关，并且该网关IP地址与其他计算机的IP地址不相同，则计算机可以连接到网络。如果计算机不能连接到任意一个默认网关，则说明网线没有插好，或网络适配器出现故障。
- 计算机的网络适配器是否出现故障？如果计算机不能连接到任意一个默认网关，则可能网络适配器出现故障，此时可以尝试对本地回环地址进行ping操作，用来确认。

此外，到DHCP、DNS、WINS的连通性也应该是关注的内容。对到服务器的连通性进行故障排除也可以采用类似的方式。如果计算机可以连接到默认网关，但不能连接到DNS、DHCP或WINS服务器，则说明服务器可能已经当机、配置的IP地址不正确，或者可能是当前操作的计算机到目标服务器的其他互相联络中断。要记住的是，你可以使用**netsh interface ipv4 show interfaces**命令与**netsh interface ipv6 show interfaces**命令，来检查网络适配器的连接状态。



附录 A

基本命令行工具参考

本书讨论了很多命令行工具与脚本，本附录提供了关于这些工具语法与用途的简明参考，也包含了很多本书未曾讲解但很有用的其他工具。且这些工具以工具名的字母先后顺序列出。除非特别提及，这些工具在Windows Server 2008与Windows Vista上的用法与工作方式是一致的。此外，如果某个工具没有同时包含在这两个版本的操作系统中，则会给出该工具的出处，比如“仅适用于Windows Server 2008”的含义是，默认情况下，该工具只在Windows Server 2008中可用。

ARP

显示与修改地址解析协议（ARP）使用的IP地址到网卡物理地址之间的转换表。

```
arp -a [inet_addr] [-N if_addr]
arp -d inet_addr [if_addr]
arp -s inet_addr eth_addr [if_addr]
```

ASSOC

显示与修改文件扩展关联。

```
assoc [.ext[=[fileType]]]
```

ATTRIB

显示与修改文件属性。

```
attrib [+r|-r] [+a|-a] [+s|-s] [+h|-h] [+i|-i]
[[drive:] [path] filename] [/s [/d] [/l]]
```

BCDEDIT

显示与管理引导配置数据（BCD）库文件。

```
bcdedit /command [options]
bcdedit [/v]
```

通过Bcdedit，可以使用如下一些命令。

- /bootdebug。激活或禁用引导应用程序的引导调试。

- **/bootems**。激活或禁用引导应用程序的应急管理服务。
- **/bootsequence**。为引导管理器设置一次性的引导顺序。
- **/copy**。复制BCD库中的条目。
- **/create**。在BCD库中创建新条目。
- **/createstore**。创建新（空）引导配置数据库文件。
- **/dbgsettings**。设置全局调试器参数。
- **/debug**。激活或禁用操作系统条目的内核调试。
- **/default**。设置引导管理器的默认条目。
- **/delete**。从BCD库中删除相应条目。
- **/deletevalue**。从BCD库中删除相应条目选项。
- **/displayorder**。设置引导管理器显示可用操作系统的顺序。
- **/ems**。为操作系统条目激活或禁用应急管理服务。
- **/emssettings**。设置全局应急管理服务参数。
- **/enum**。列出库中的条目。
- **/export**。将系统库中的内容导出到文件，该文件可用于恢复操作系统库的状态。
- **/import**。使用由/export命令创建的备份文件恢复操作系统库的状态。
- **/set**。设置BCD库中的条目选项值。
- **/timeout**。设置引导管理器的超时值。
- **/toolsdisplayorder**。设置引导管理器显示工具菜单的顺序。

CACLS

该命令已经过时，参见ICACLS。

CALL

以过程的形式调用脚本或脚本标记。

```
call [drive:][path] filename [batch-parameters]
call :label [args]
```

CD

显示目录名或切换当前目录。

```
chdir [/d] [drive:][path]
chdir [..]
cd [/d] [drive:][path]
cd [..]
```

CHDIR

参见CD。

CHKDSK

检查磁盘的错误并显示报告。

```
chkdsk [drive:][[path] filename]
[/f][/v][/r][/x][/i][/c][/l[:size]] [/b]
```

CHKNTFS

显示卷状态，在计算机启动时设置或排除一些系统自动检测过程中涉及到的卷。

```
chkntfs [/x | /c] volume: [...]
chkntfs /t[:time]
chkntfs /d
```

CHOICE

创建一个选择列表，通过该列表，用户可以在批处理脚本中选择相应的选项。

```
choice [/c choices] [/n] [/cs] [/t nnnn /d choice] [/m "text"]
```

CIPHER

显示当前的加密状态，或在NTFS卷上修改文件夹与文件加密。

```
cipher [/e | /d | /c] [/s:dir] [/b]
[/h] [[path]filename [...]]
cipher [/k | /r:filename | /w:dir]
cipher /u [/n]
cipher /x[:efsfile] [filename]
cipher /y
cipher /adduser [/certhash:hash | /certfile:filename]
[/s:dir] [/b] [/h] [pathname [...]]
cipher /removeuser /certhash:hash
[/s:dir] [/b] [/h] [pathname [...]]
cipher /rekey [pathname [...]]
```

CLIP

与管道技术类似，键命令行工具的输出重定向到写字板。

```
[command |] clip
Clip < filename.txt
```

注解 本实例中，符号| 为管道符号。

CLS

清空控制台窗口。

```
cls
```

CMD

启动一个新的命令shell实例。

```
cmd [/a | /u] [/q] [/d] [/e:on | /e:off] [/t:{bf | f}]  
[/f:on | /f:off] [/v:on | /v:off]  
[[/s] [/c | /k] string]
```

CMDKEY

创建与管理存储的用户名与口令。

```
cmdkey [{/add | /generic}:targetname  
{/smartcard | /user:user@domain  
{/pass:{pwd}} | /delete{:targetname}  
| /ras | /list{:targetname}]
```

COLOR

设置命令shell窗口的颜色。

```
color [[b] f]
```

COMP

比较两个文件或文件集的内容。

```
comp [data1] [data2] [/d] [/a] [/l]  
[/n=number] [/c] [/offline]
```

COMPACT

在NTFS分区上显示或修改文件压缩。

```
compact [/c | /u] [/s[:dir]] [/a] [/i] [/f]  
[/q] [filename [...]]
```

CONVERT

将FAT或FAT32卷转换为NTFS格式。

```
convert volume /fs:NTFS [/v] [/x]  
[/cvtarea:filename] [/nosecurity]
```

COPY

复制或合并文件。

```
copy [/d] [/v] [/n] [/y|/y] [/z] [/l] [/a|/b] source [/a | /b]
[+ source [/a | /b] [+ ...]][destination [/a | /b]]
```

DATE

显示或设置系统日期。

```
date [/T | mm-dd-yy]
```

DCDIAG

在域控制器上执行诊断性测试。

```
dcdiag [/s:Server[:LDAPPort]] [/u [Domain\]UserName]
[/p {Password | * | ""}] [/h | /?] [/xs] [/a | /e] [/I] [/fix] [/c]
[/q | /v] [/n:NamingContext] [/skip:TestName] [/test:TestName]
[/f:textlogname] [/x:xmllogname]
```

该命令仅适用于Windows Server 2008。

DCGPOFIX

恢复默认的组策略对象。

```
dcgpofix [/ignore schema]
[/target: {domain | dc | both}]
```

该命令仅适用于Windows Server 2008。

DEFRAG

对硬盘驱动器进行碎片整理。

```
defrag volume [/a] [/v]
defrag [volume | -c] [{-r | -w}] [-f] [-v]
```

只能在管理员命令提示符下运行该命令。

DEL

删除一个或多个文件。

```
del [/p] [/f] [/s] [/q] [/a[:attributes]]
[drive:][path]filename[...]
```

DIR

显示目录内的文件与子目录列表。

```
dir [drive:][path][filename] [/a[:attributes]] [/b] [/c] [/d]
[/l] [/n] [/o[:sortorder]] [/p] [/q] [/r] [/s] [/t[:timefield]]
[/w] [/x] [/4]
```

DISKCOMP

比较两个软盘的内容。

```
diskcomp [drive1: [drive2:]]
```

DISKCOPY

将一个软盘的内容复制到另外的软盘。

```
diskcopy [drive1: [drive2:]] [/v]
```

DISKPART

调用一个文本模式的命令解释器，来使用单独的命令提示符管理磁盘、分区与卷，以及DISKPART的内部命令。

```
diskpart
```

更多信息 第10章、第11章、第12章都有部分内容讲解了如何使用DISKPART。如果在非管理员权限的命令提示符中运行DISKPART，会收到提示信息，声称DISKPART必须在管理员权限的命令提示符中运行。用于Windows Vista SP1与Windows Server 2008 RTM的DISKPART对Windows Vista RTM中的DISKPART进行了修订，且修订中包含SAN与UNIQUEID两个附加的命令。

DOSKEY

编辑命令行，重新调用命令行，创建宏。

```
doskey [/reinstall] [/listsize=size]  
[/macros[:all | :exename]]  
[/history] [/insert | /overstrike]  
[/exename=exename]  
[/macrofile=fname] [macroname=[text]]
```

DRIVERQUERY

显示所有已安装设备驱动程序及其属性列表。

```
driverquery [/s computer [/u [domain\]user [/p [pwd]]]]  
[/fo {table|list|csv}} [/nh] [/v] [/si]
```

DSADD COMPUTER

在活动目录目录服务中创建一个计算机账号。

```
dsadd computer ComputerDN [-samid SAMName] [-desc Description]  
[-loc Location] [-memberof GroupDN ...] [{-s Server | -d Domain}]  
[-u UserName] [-p {Password | *}] [-q] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSADD GROUP

在活动目录中创建一个组账号。

```
dsadd group GroupDN [-secgrp {yes | no}] [-scope {l | g | u}]
[-samid SAMName] [-desc Description] [-memberof Group ...]
[-members Member ...] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-q] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSADD USER

在活动目录中创建一个用户。

```
dsadd user UserDN [-samid SAMName] [-upn UPN] [-fn FirstName]
[-mi Initial] [-ln LastName] [-display DisplayName]
[-empid EmployeeID] [-pwd {Password | *}] [-desc Description]
[-memberof Group ...] [-office Office] [-tel PhoneNumber]
[-email EmailAddress] [-hometel HomePhoneNumber]
[-pager PagerNumber] [-mobile CellPhoneNumber] [-fax FaxNumber]
[-iptel IPPhoneNumber] [-webpg WebPage] [-title Title]
[-dept Department] [-company Company] [-mgr Manager]
[-hmdir HomeDirectory] [-hmdrv DriveLetter:] [-profile ProfilePath]
[-loscr ScriptPath] [-mustchpwd {yes | no}] [-canchpwd {yes | no}]
[-reversiblepwd {yes | no}] [-pwdneverexpires {yes | no}]
[-acctexpires NumberOfDays] [-disabled {yes | no}]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}]
[-q] [{-uc | -uco | -uci}] [-fnp FirstNamePhonetic]
[-lnp LastNamePhonetic] [-displayp DisplayNamePhonetic]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSGET COMPUTER

使用两种语法格式显示计算机账号的属性, 查看多台计算机属性的语法格式为:

```
dsget computer ComputerDN ... [-dn] [-samid] [-sid] [-desc] [-loc]
[-disabled] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
[-part PartitionDN [-qlimit] [-qued]]
```

查看一个计算机成员信息的语法格式为:

```
dsget computer ComputerDN [-memberof [-expand]]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-c]
[-q] [-l] [{-uc | -uco | -uci}]
```


如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSGET GROUP

使用两种语法格式显示组账号的属性, 查看多个组属性的语法格式为:

```
dsget group GroupDN ... [-dn] [-samid] [-sid] [-desc] [-secgrp]
[-scope] [{-s Server | -d Domain}] [-u UserName] [-p {Password | *}]
[-c] [-q] [-l] [{-uc | -uco | -uci}] [-part PartitionDN [-qlimit]
[-qused]]
```

查看一个组成员信息的语法格式为:

```
dsget group GroupDN [{-memberof | -members} [-expand]]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}]
[-c] [-q] [-l] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSGET SERVER

使用3种语法格式显示域控制器的属性, 显示指定域控制器常规属性的语法格式为:

```
dsget server ServerDN ... [-dn] [-desc] [-dnsname] [-site]
[-isgc] [{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-c]
[-q] [-l] [{-uc | -uco | -uci}]
```

显示安全主体(在指定域控制器上拥有最大数量目录对象)列表的语法格式为:

```
dsget server ServerDN [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
[-topobjowner NumbertoDisplay]
```

显示指定域控制器上目录分区区分名的语法格式为:

```
dsget server ServerDN [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}] [-part]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSGET USER

使用两种语法格式显示用户账号的属性, 查看多个用户属性的语法格式为:

```
dsget user UserDN ... [-dn] [-samid] [-sid] [-upn] [-fn] [-mi]
[-ln] [-display] [-fnp] [-lnp] [-displayp] [-effectivepso]
[-empid] [-desc] [-office] [-tel] [-email] [-hometel] [-pager]
[-mobile] [-fax] [-iptel] [-webpg] [-title] [-dept] [-company]
[-mgr] [-hmdir] [-hmdrv] [-profile] [-loscr] [-mustchpwd] [-canchpwd]
```

```
[-pwdneverexpires] [-disabled] [-acctexpires] [-reversiblepwd]
[{-uc | -uco | -uci}] [-part PartitionDN [-qlimit] [-qused]]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-c] [-q]
[-l]
```

查看用户组成员关系信息的语法格式为：

```
dsget user UserDN [-memberof [-expand]] [{-uc | -uco | -uci}]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-c]
[-q] [-l]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSMGMT

调用一个文本模式的命令解释器，以便使用单独的命令提示符管理目录服务以及DSMGMT的内部命令。

```
dsmgmt
```

DSMOD COMPUTER

修改目录中一个或多个计算机账号的属性。

```
dsmod computer ComputerDN ... [-desc Description] [-loc Location]
[-disabled {yes | no}] [-reset] [{-s Server | -d Domain}] [-u UserName]
[-p {Password | *}] [-c] [-q] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSMOD GROUP

修改目录中一个或多个组账号的属性。

```
dsmod group GroupDN ... [-samid SAMName] [-desc Description]
[-secgrp {yes | no}] [-scope {l | g | u}]
[{-addmbr | -rmmbr | -chmbr} MemberDN ...] [{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-c] [-q] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSMOD SERVER

修改域控制器的属性。

```
dsmod server ServerDN ... [-desc Description] [-isgc {yes | no}]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-c]
[-q] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSMOD USER

修改目录中一个或多个用户账号的属性。

```
dsmod user UserDN ... [-upn UPN] [-fn FirstName] [-mi Initial]
[-ln LastName] [-display DisplayName] [-empid EmployeeID]
[-pwd {Password | *}] [-desc Description] [-office Office]
[-tel PhoneNumber] [-email EmailAddress] [-hometel HomePhoneNUmber]
[-pager PagerNumber] [-mobile CellPhoneNUmber] [-fax FaxNumber]
[-iptel IPPhoneNumber] [-webpg WebPage] [-title Title]
[-dept Department] [-company Company] [-mgr Manager]
[-hmdir HomeDirectory] [-hmdrv DriveLetter:] [-profile ProfilePath]
[-loscr ScriptPath] [-mustchpwd {yes | no}] [-canchpwd {yes | no}]
[-reversiblepwd {yes | no}] [-pwdneverexpires {yes | no}]
[-acctexpires NumberOfDays] [-disabled {yes | no}]
[{-s Server | -d Domain}] [-u UserName] [-p {Password | *}] [-c] [-q]
[{-uc | -uco | -uci}] [-fnp FirstNamePhonetic]
[-lnp LastNamePhonetic] [-displayp DisplayNamePhonetic]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSMOVE

移动活动目录对象或对其重命名。

```
dsmove objectdn [-newname newname] [-newparent parentdn]
[{-s server | -d domain}] [-u username] [-p {password | *}] [-q]
[{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY COMPUTER

搜索匹配特定标准的计算机账号。

```
dsquery computer [{startnode | forestroot | domainroot}]
[-o {dn | rdn | samid}] [-scope {subtree | onelevel | base}] [-name name]
[-desc description] [-samid samname] [-inactive numberofweeks]
[-stalepwd numberofdays] [-disabled] [{-s server | -d domain}]
[-u username] [-p {password | *}] [-q] [-r] [-gc]
[-limit numberofobjects]
[{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装, 则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY CONTACT

搜索匹配特定标准的联系人。

```
dsquery contact [{startnode | forestroot | domainroot}]
[-o {dn | rdn}] [-scope {subtree | onelevel | base}] [-name name]
[-desc description] [{-s server | -d domain}] [-u username]
[-p {password | *}] [-q] [-r] [-gc] [-limit numberofobjects]
[{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY GROUP

搜索匹配特定标准的组账号。

```
dsquery group [{startnode | forestroot | domainroot}]
[-o {dn | rdn | samid}] [-scope {subtree | onelevel | base}]
[-name name] [-desc description] [-samid SAMName]
[{-s server | -d domain}] [-u username] [-p {password | *}] [-q]
[-r] [-gc] [-limit numberofobjects] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY PARTITION

搜索匹配特定标准的活动目录分区。

```
dsquery partition [-o {dn | rdn}] [-part filter] [-desc description]
[{-s server | -d domain}] [-u username] [-p {password | *}] [-q]
[-r] [-limit numberofobjects] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY QUOTA

搜索匹配特定标准的磁盘配额。

```
dsquery quota {domainroot | objectdn} [-o {dn | rdn}] [-acct name]
[-qlimit filter] [-desc description] [{-s server | -d domain}]
[-u username] [-p {password | *}] [-q] [-r] [-limit numberofobjects]
[{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY SERVER

搜索匹配特定标准的域控制器。

```
dsquery server [-o {dn | rdn}] [-forest] [-domain domainname]
[-site sitename] [-name name] [-desc description]
[-hasfsmo {schema | name | infr | pdc | rid}] [-isgc]
[-isreadonly] [{-s server | -d domain}] [-u username]
[-p {password | *}] [-q] [-r] [-gc] [-limit numberofobjects]
[{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY SITE

搜索匹配特定标准的活动目录站点。

```
dsquery site [-o {dn | rdn}] [-name name] [-desc description]
[{-s server | -d domain}] [-u username] [-p {password | *}] [-q]
[-r] [-gc] [-limit numberofobjects] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY USER

搜索匹配特定标准的用户账号。

```
dsquery user [{startnode | forestroot | domainroot}]
[-o {dn | rdn | upn | samid}] [-scope {subtree | onelevel | base}]
[-name name] [-namep namephonetic] [-desc description] [-upn upn]
[-samid samname] [-inactive numberofweeks] [-stalepwd numberofdays]
[-disabled] [{-s server | -d domain}] [-u username]
[-p {password | *}] [-q] [-r] [-gc] [-limit numberofobjects]
[{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSQUERY *

搜索匹配特定标准的任意活动目录对象。

```
dsquery * [{startnode | forestroot | domainroot}]
[-scope {subtree | onelevel | base}] [-filter ldapfilter]
[-attr {attributelist | *}] [-attrsonly] [-l]
[{-s server | -d domain}] [-u username] [-p {password | *}] [-q]
[-r] [-gc] [-limit numberofobjects] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

DSRM

删除活动目录对象。

```
dsrm objectdn ... [-subtree [-exclude]] [-noprompt]
[{-s server | -d domain}] [-u username] [-p {password | *}] [-c]
[-q] [{-uc | -uco | -uci}]
```

如果Windows Server 2008管理工具包已经安装，则该命令在Windows Vista Business及其后续版本中是可用的。

ECHO

显示消息，或者打开/关闭命令回显。

```
echo [on | off]
echo [message]
```

ENDLOCAL

在批文件中终止环境变量的局部化。

```
endlocal
```

ERASE

参见DEL。

ESENTUTL

管理可扩展存储引擎（ESE）数据库，包括活动目录域服务（ADDS）使用的。

碎片整理语法：

```
esentutl /d databasename /s [streamingfilename] /t [tempdbname]
/f [tempstreamingfilename] /i /p /b [backupfilename] /8 /o
```

恢复语法：

```
esentutl /r logfilebasename /l [logdirectory]
/s [systemfilesdirectory]
/i /t /u [log] /d [dbfiledirectory] /n path1[:path2] /8 /o
```

完整性检查语法：

```
esentutl /g databasename /s [streamingfilename] /t [tempdbname]
/f [tempstreamingfilename] /i /8 /o
```

校验和计算语法：

```
esentutl /k filetocheck /s [streamingfilename]
/t [tempdbname] /p nn
/e /i /8 /o
```


修复语法:

```
esentutl /p databasename /s [streamingfilename] /t [tempdbname]
/f [reportprefix] /i /g /createstm /8 /o
```

转储文件语法:

```
esentutl /m [h|k|l|m|s|u] filename /p pagenumber
/s [streamingfilename] /t tablename /v /8 /o
```

复制文件语法:

```
esentutl /y sourcefile /d destinationfile /o
```

EVENTCREATE

在事件日志中创建自定义事件。

```
eventcreate [/s computer [/u domain\user [/p password]]
[/l {application | system}] | [/so srcname]
/t {success | error | warning | information} /id eventid
/d description
```

EXIT

退出命令解释器。

```
exit [/b] [exitcode]
```

EXPAND

取消文件的压缩。

```
expand [-r] source destination
expand -r source [destination]
expand -d source.cab [-f:files]
expand source.cab -f:files destination
```

FC

比较文件并显示差别。

```
fc [/a] [/c] [/l] [/lbn] [/n] [/t] [/u] [/w]
[/nnnn] [/offline] [drive1:][path1] filename1
[drive2:][path2] filename2
fc /b [drive1:][path1] filename1
[drive2:][path2] filename2
```

FIND

在文件中搜索文本字符串。

```
find [/v] [/c] [/n] [/i] [/offline] "string"
[[drive:][path]filename[ ...]]
```

FINDSTR

使用正则表达式在文件中搜索字符串。

```
findstr [/b] [/e] [/l] [/r] [/s] [/i] [/x] [/v] [/n]
[/m] [/o] [/p] [/f:file] [/a:attr] [/c:string]
[/d:dir] [/g:file] [/offline] [strings]
[[drive:][path]filename[ ...]]
```

FOR

对文件集中的每一文件运行指定命令。

命令行FOR循环：

```
for %variable in (set) do command [parameters]
for /d %variable in (set) do command [parameters]
for /r [[drive:]path] %variable in (set) do command [parameters]
for /l %variable in (start,step,end) do command [parameters]
for /f ["options"] %variable in (set) do command [parameters]
```

脚本FOR循环：

```
for %%variable in (set) do command [parameters]
for /d %%variable in (set) do command [parameters]
for /r [[drive:]path] %%variable in (set) do command [parameters]
for /l %%variable in (start,step,end) do command [parameters]
for /f ["options"] %%variable in (set) do command [parameters]
```

FORFILES

选择一个或多个文件并对每一文件执行命令。

```
forfiles [/p pathname] [/m searchmask] [/s] [/c command]
[/d [+ | -] {mm/dd/yyyy | dd}]
```

FORMAT

对软盘或硬盘进行格式化。

```
format drive: [/fs:file-system] [/v:label] [/q] [/a:size] [/c]
[/x] [/p:numzerofillpasses]
format drive: [/v:label] [/q] [/f:size | /t:tracks /n:sectors]
[/p:numzerofillpasses]
```

FTP

传输文件。

```
ftp [-v] [-d] [-i] [-n] [-g] [-s:filename] [-a] [-A] [-x:sendbuffer]
[-r:recvbuffer] [-b:asyncbuffers] [-w:window size] [host]
```

用于FTP命令的参数是区分大小写，注意要以上面语法格式中的大小写格式进行输入。

FTYPE

显示或修改文件扩展关联中使用的文件类型。

```
ftype [fileType]=[command]]
```

GET-EVENTLOG

Windows PowerShell命令，用于显示事件日志的信息或存储在事件日志中的条目。

```
get-eventlog -list
get-eventlog [-logname] logname [-newest nn]
```

GET-PROCESS

Windows PowerShell命令，用于显示运行中进程信息。

```
get-process -id [id1,id2,...]
get-process -inputobject processname1, processname2, ... [process ...]
get-process [-name] [processname1, processname2,...]
```

GET-SERVICE

Windows PowerShell命令，用于显示已配置服务信息。

```
get-service [-displayname [servicename1, servicename2,...]]
[-include [servicename1, servicename2,...]]
[-exclude [servicename1, servicename2,...]]
get-service [-name] [servicename1, servicename2,...]
[-include [servicename1, servicename2,...]]
[-exclude [servicename1, servicename2,...]]
get-service [-inputobject servicename1, servicename2,...]
[-include [servicename1, servicename2,...]]
[-exclude [servicename1, servicename2,...]]
```

GETMAC

显示网络适配器信息。

```
getmac [/s computer [/u [domain]\user [/p [pwd]]]]
[/fo {table|list|csv}] [/nh] [/v]
```

GOTO

跳转到脚本中标号行。

```
goto :label  
goto :EOF
```

GPUPDATE

强制进行组策略的后台刷新。

```
gpupdate [/target:{computer | user}] [/force] [/wait:<value>]  
[/logoff] [/boot] [/sync]
```

HOSTNAME

打印计算机名。

```
hostname
```

ICACLS

显示或修改文件的访问控制列表（ACL）。

将所有匹配名的ACL存储到ACL文件，语法格式如下：

```
icacls name /save aclfile [/t] [/c] [/l] [/q]
```

将存储的ACL恢复到目录中的某个文件，语法格式如下：

```
icacls directory [/substitute sidold sidnew [...]] /restore aclfile  
[/c] [/l] [/q]
```

改变所有匹配名的属主，语法格式如下：

```
icacls name /setowner user [/t] [/c] [/l] [/q]
```

使用特定SID寻找所有匹配名，语法格式如下：

```
icacls name /findsid sid [/t] [/c] [/l] [/q]
```

授予许可权限，语法格式如下：

```
icacls name [/grant[:r] sid:perm [...]]
```

取消许可权限，语法格式如下：

```
icacls name [/deny sid:perm [...]]
```

移除许可权限，语法格式如下：

```
icacls name [/remove[:g]:d] sid[...] [/t] [/c] [/l] [/q]
```

将ACL重置为继承值，语法如下：

```
icacls name /reset [/t] [/c] [/l] [/q]
```

设置完整性级别，语法格式如下：

```
icacls name [/setintegritylevel level:policy[...]]
```

验证ACL，语法格式如下：

```
icacls name /verify [/t] [/c] [/l] [/q]
```

IF

在批处理脚本程序中进行条件处理。

```
if [not] errorlevel number command
if [not] [/i] string1==string2 command
if [not] exist filename command
if [/i] string1 compare-op string2 command
if cmdextversion number command
if defined variable command
```

IPCONFIG

显示TCP/IP配置。

```
ipconfig [/allcompartments] {/all}
ipconfig [/release [adapter] | /renew [adapter]
        | /release6 [adapter] | /renew6 [adapter]]
ipconfig /flushdns | /displaydns | /registerdns
ipconfig /showclassid adapter
ipconfig /setclassid adapter [classidtoreset]
```

LABEL

创建、改变或删除磁盘的卷标。

```
label [drive:][label]
label [/mp] [volume] [label]
```

MD

创建目录或子目录。

```
mkdir [drive:]path
md [drive:]path
```

MKDIR

参见MD。

MORE

输出一次显示在一个屏幕。

```
more [/e [/c] [/p] [/s] [/tn] [+n]] < [drive:][path]filename  
more /e [/c] [/p] [/s] [/tn] [+n] [files]  
command-name | more [/e [/c] [/p] [/s] [/tn] [+n]]
```

MOUNTVOL

管理卷挂载点。

```
mountvol [drive:]path volumeName  
mountvol [drive:]path {/d | /l | /p}  
mountvol [/r | /n | /e]
```

MOVE

将文件从某个目录移动到同一驱动器上的其他目录。

```
move [/y] [/-y] source target
```

NBTSTAT

显示NETBIOS状态。

```
nbtstat [-a remotename] [-A ipaddress] [-c] [-n] [-r] [-R] [-RR]  
[-s] [-S [interval]]
```

注解 该命令使用区分大小写的参数。

NET ACCOUNTS

管理用户账号与口令策略。

```
net accounts [/forceloff:{minutes | no}]  
[/minpwlen:length]  
[/maxpwage:{days | unlimited}]  
[/minpwage:days]  
[/uniquepw:number] [/domain]
```

NET COMPUTER

从域中移除计算机或向域内添加计算机。

```
net computer \\computername {/add | /del}
```

NET CONFIG SERVER

显示或修改server服务的配置。

```
net config server [/autodisconnect:time]  
[/srvcomment:"text"] [/hidden:{yes | no}]
```


NET CONFIG WORKSTATION

显示或修改workstation服务的配置。

```
net config workstation [/charcount:bytes]
[/chartime:msec]
[/charwait:sec]
```

NET CONTINUE

恢复暂停的服务。

```
net continue service
```

NET FILE

显示或管理服务器上的打开文件。

```
net file [id [/close]]
```

NET GROUP

显示或管理全局组。

```
net group [groupname [/comment:"text"]]
[/domain]
net group groupname {/add [/comment:"text"]
| /delete} [/domain]
net group groupname username [...]
{/add | /delete} [/domain]
```

NET LOCALGROUP

显示本地组账号。

```
net localgroup [GroupName [/comment:"Text"]] [/domain]
```

创建本地组账号。

```
net localgroup GroupName {/add [/comment:"Text"]} [/domain]
```

修改本地组账号。

```
net localgroup [GroupName Name [ ...] /add [/domain]
```

删除本地组账号。

```
net localgroup GroupName /delete [/domain]
```

NET PAUSE

挂起服务。

```
net pause service
```

NET PRINT

显示或管理打印任务与共享队列。

```
net print \\computername\sharename  
net print [\\computername] job# [/hold | /release | /delete]
```

NET SESSION

列出或断开会话。

```
net session [\\computername] [/delete]
```

NET SHARE

显示或管理共享的打印机与目录。

```
net share [sharename]  
net share sharename[=drive:path] [/users:number | /unlimited]  
    [/remark:"text"] [/cache:flag]  
net share {sharename | devicename | drive:path} /delete
```

NET START

列出或启动网络服务。

```
net start [service]
```

NET STATISTICS

显示工作站与服务器统计信息。

```
net statistics [workstation | server]
```

NET STOP

终止服务。

```
net stop service
```

NET TIME

显示或同步网络时间。

```
net time [\\computername | /domain[:domainname] |  
    /rtdomain[:domainname]] [/set]  
net time [\\computername] /querysnTP  
net time [\\computername] /setsntp[:serverlist]
```

NET USE

显示或管理远程连接。

```
net use [devicename | *] [\\computername\sharename[\volume]
[password | *]] [/user:[domainname\]username]
[/user:[username@domainname]] [[/delete] | [/persistent:{yes | no}]]
[/smartcard] [/savecred]
net use [devicename | *] [password | *]] [/home]
net use [/persistent:{yes | no}]
```

NET USER

创建本地用户账号。

```
net user UserName [Password | *] /add [/active:{no | yes}]
[/comment:"DescriptionText"] [/countrycode:NNN]
[/expires: {{MM/DD/YYYY | DD/MM/YYYY | mmm,dd,YYYY} | never}]
[/fullname:"Name"] [/homedir:Path] [/passwordchg:{yes | no}]
[/passwordreq:{yes | no}] [/profilepath:[Path]] [/scriptpath:Path]
[/times:{Day[-Day][,Day[-Day]] ,Time[-Time][,Time[-Time]]
[;...] | all}] [/usercomment:"Text"]
[/workstations:{ComputerName[,...] | *}] [/domain]
```

修改本地用户账号。

```
net user [UserName [Password | *] [/active:{no | yes}]
[/comment:"DescriptionText"] [/countrycode:NNN]
[/expires: {{MM/DD/YYYY | DD/MM/YYYY | mmm,dd,YYYY} | never}]
[/fullname:"Name"] [/homedir:Path] [/passwordchg:{yes | no}]
[/passwordreq:{yes | no}] [/profilepath:[Path]] [/scriptpath:Path]
[/times:{Day[-Day][,Day[-Day]] ,Time[-Time][,Time[-Time]]
[;...] | all}] [/usercomment:"Text"]
[/workstations:{ComputerName[,...] | *}] [/domain]
```

删除本地用户账号。

```
net user UserName [/delete] [/domain]
```

NET VIEW

显示网络资源或计算机。

```
net view [\\computername [/cache] | [/all] |
/domain[:domainname]]
net view /network:nw [\\computername]
```

NETDOM ADD

向域内添加一个工作站或服务器账号。

```
netdom add computer [/domain:domain] [/userd:user]
[/passwordd:[password | *]]
[/server:server] [/ou:oupath] [/dc] [/securepasswordprompt]
```

NETDOM COMPUTENAME

管理计算机的主要名与备用名，该命令可以安全地对域控制器或服务器进行重命名。

```
netdom computename computer [/usero:user]
[/password:[password | *]]
[/userd:user] [/passwordd:[password | *]] [/securepasswordprompt]
/add:newalternatednsname | /remove:alternatednsname |
/makeprimary:computerdnsname |
/enumerate[:{alternatenames | primaryname | allnames}] | /verify
```

NETDOM JOIN

将工作站或成员服务器添加到域内。

```
netdom join computer /domain:domain [/ou:oupath] [/userd:user]
[/passwordd:[password | *]]
[/usero:user] [/passwordo:[password | *]]
[/reboot[:timeinseconds]]
[/securepasswordprompt]
```

NETDOM MOVE

将工作站或成员服务器移动到新域内。

```
netdom move computer /domain:domain [/ou:oupath]
[/userd:user] [/passwordd:[password | *]]
[/usero:user] [/passwordo:[password | *]]
[/userf:user] [/passwordf:[password | *]]
[/reboot[:timeinseconds]]
[/securepasswordprompt]
```

NETDOM MOVENT4BDC

将Windows NT 4.0备份域控制器移动到新域内。

```
netdom movent4bdc computer [/domain:domain] [/reboot[:timeinseconds]]
```

NETDOM QUERY

查询某个域的相关信息。

```
netdom query [/domain:domain] [/server:server]
[/userd:user] [/passwordd:[password | *]]
[/verify] [/reset] [/direct] [/securepasswordprompt]
{workstation | server | dc | ou | pdc | fsmo | trust}
```

NETDOM REMOVE

从域中移除工作站或服务器。

```
netdom remove computer [/domain:domain] [/userd:user]
[/passwordd:[password | *]]
[/usero:user] [/passwordo:[password | *]]
[/reboot[:timeinseconds]] [/force]
[/securepasswordprompt]
```

NETDOM RENAMECOMPUTER

对计算机进行重命名。如果计算机加入到域中，则域中相应的计算机对象也被重命名。要注意的是，不能使用本命令对域控制器进行重命名。

```
netdom renamecomputer computer /newname:newname
[/userd:user [/passwordd:[password | *]]]
[/usero:user [/passwordo:[password | *]]]
[/force] [/reboot[:timeinseconds]]
[/securepasswordprompt]
```

NETDOM RESET

重置工作站与域控制器之间的安全连接。

```
netdom reset computer [/domain:domain] [/server:server]
[/usero:user] [/passwordo:[password | *]] [/securepasswordprompt]
```

NETDOM RESETPWD

重置本命令所运行域控制器上的机器账号口令。

```
netdom resetpwd /server:domaincontroller /userd:user
/passwordd:[password | *]
[/securepasswordprompt]
```

NETDOM TRUST

管理或验证域间的信任关系。

```
netdom trust trustingdomainname /domain:trusteddomainname [/userd:user]
[/passwordd:[password | *]] [/usero:user] [/passwordo:[password | *]]
[/verify] [/reset] [/passwordt:newrealmtrustpassword]
[/add] [/remove] [/twoway] [/realm] [/kerberos]
[/transitive[:{yes | no}]]
[/oneside:{trusted | trusting}] [/force] [/quarantine[:{yes | no}]]
[/namesuffixes:trustname [/togglesuffix:#]]
[/enablesidhistory[:{yes | no}]]
[/foresttransitive[:{yes | no}]]
[/crossorganization[:{yes | no}]]
[/addtln:toplevelname]
[/addtl nex:toplevelnameexclusion]
[/removetln:toplevelname]
[/removetl nex:toplevelnameexclusion]
[/securepasswordprompt]
```

NETDOM VERIFY

验证工作站与域控制器之间的安全连接。

```
netdom verify computer [/domain:domain] [/usero:user]  
[/password:[password | *]] [/securepasswordprompt]
```

NETSH

调用一个单独的命令提示符，用于对本地计算机或远程计算机上的网络服务配置进行管理。

```
netsh
```

更多信息 Netsh的操作技术在第17章有详细的讲解。

NETSTAT

显示网络连接的状态。

```
netstat [-a] [-b] [-e] [-f] [-n] [-o] [-p protocol] [-r] [-s] [-t]  
[interval]
```

NSLOOKUP

显示DNS状态。

```
nslookup [-option] [computer]  
nslookup [-option] [computer server]
```

PATH

在当前命令窗口中，显示或设置可执行文件的搜索路径。

```
path [[drive:]path[;...]][%PATH%]  
path ;
```

PATHPING

对路由进行追踪并提供丢包信息。

```
pathping [-n] [-h maxhops] [-g hostlist]  
[-i address] [-p period]  
[-q numqueries] [-w timeout]  
targetname [-4] [-6]
```

PAUSE

挂起对脚本的处理，直至有键盘输入。


```
pause
```

PING

确定是否可以建立网络连接。

```
ping [-t] [-a] [-n count] [-l size] [-f]
      [-i ttl] [-v tos] [-r count] [-s count]
      [[-j hostlist] | [-k hostlist]]
      [-w timeout] [-R] [-S sourceaddress]
      [-4] [-6] destinationlist
```

注解 该命令使用区分大小写的参数。

POPD

切换到由PUSHHD保存的目录。

```
popd
```

PRINT

打印文本文件。

```
print [/d:device]
      [[drive:][path] filename[...]]
```

PROMPT

改变Windows命令提示符。

```
prompt [text]
```

PUSHD

保存当前目录，之后切换到新目录。

```
pushd [path | ..]
```

RD

移除目录。

```
rmdir [/s] [/q] [drive:]path
rd [/s] [/q] [drive:]path
```

RECOVER

从损坏的或有缺陷的磁盘中恢复可读信息。

```
recover [drive:][path]filename
```

REG ADD

向注册表中添加一个子键或条目。

```
reg add keyname [/v valuenam | /ve] [/t datatype] [/d data] [/f]  
[/s separator]
```

REG COMPARE

比较注册表子键或条目。

```
reg compare keyname1 keyname2 [/v valuenam | /ve] [/s]  
[/outputoption]
```

REG COPY

将注册表条目复制到本地或远程系统上的特定键路径。

```
reg copy keyname1 keyname2 [/s] [/f]
```

REG DELETE

从注册表中删除一个子键或条目。

```
reg delete keyname [/v valuenam | /ve | /va] [/f]
```

REG QUERY

列出某注册表键下的条目与子键名（如果有）。

```
reg query keyname [/v valuenam | /ve] [/s]  
[/f data [/k] [/d] [/c] [/e]] [/t type] [/z] [/se separator]
```

REG RESTORE

将保存的子键与条目写回到注册表。

```
reg restore keyname "filename"
```

REG SAVE

将指定子键、条目、值的副本保存到某个文件中。

```
reg save keyname "filename" [/y]
```

REGSVR32

注册或取消对DLL的注册。

```
regsvr32 [/u] [/s] [/n] [/i[:cmdline]] dllname
```

REM

向脚本中添加注释。

```
rem [comment]
```

REN

对文件进行重命名。

```
rename [drive:][path]filename1 filename2  
ren [drive:][path]filename1 filename2
```

RMDIR

参见RD。

ROUTE

管理网络路由表。

```
route [-f] [-p] [-4|-6] command [destination]  
[mask netmask] [gateway] [metric metric] [if interface]
```

RUNAS

以特定用户许可权限运行程序。

以指定用户的凭据运行，语法如下：

```
runas [/noprofile | /profile] [/env] [/netonly] [/savecred]  
/user:account program
```

以来自智能卡的凭据运行，语法如下：

```
runas [/noprofile | /profile] [/env] [/netonly] [/savecred]  
/smartcard [/user:account] program
```

显示可用的信任级别，语法如下：

```
runas /showtrustlevels
```

在指定的信任级别上运行，语法如下：

```
runas /trustlevel:trustlevel program
```

SC CONFIG

配置服务的启动与登录账号。

```
sc [\\ServerName] config ServiceName
    [type= {own | share | {interact type = {own | share}} | kernel |
    filesystem | rec | adapt}]
    [start= {boot | system | auto | demand | disabled | delayed-auto}]
    [error= {normal | severe | critical | ignore}]
    [binPath= BinaryPathName]
    [group= LoadOrderGroup]
    [tag= {yes | no}]
    [depend= Dependencies]
    [obj= {AccountName | ObjectName}]
    [DisplayName= displayname]
    [password= password]
```

SC CONTINUE

恢复已暂停的服务。

```
sc [\\ServerName] continue ServiceName
```

SC FAILURE

查看服务失败时要采取的操作。

```
sc [\\ServerName] failure ServiceName [reset= ErrorFreePeriod]
    [reboot= BroadcastMessage] [command= CommandLine]
    [actions= FailureActionsAndDelayTime]
```

SC PAUSE

暂停服务。

```
sc [\\ServerName] pause ServiceName
```

SC QC

显示指定服务的配置信息。

```
sc [\\ServerName] qc ServiceName [BufferSize]
```

SC.QFAILURE

设置服务失败时采取的操作。

```
sc [\\ServerName] qfailure ServiceName [BufferSize]
```

SC QUERY

显示计算机上配置的服务列表。

```
sc [\\ServerName] query ServiceName
    [type= {driver | service | all}]
    [type= {own|share|interact|kernel|filesystem|rec|adapt}]
    [state= {active | inactive | all}] [bufsize= BufferSize]
    [ri= ResumeIndex]
    [group= GroupName]
```

SC START

启动服务。

```
sc [\\ServerName] start ServiceName [ServicesArgs]
```

SC STOP

终止服务。

```
sc [\\ServerName] stop ServiceName
```

SCHTASKS/CHANGE

改变现有任务的属性。

```
schtasks /change /tn taskname [/s system [/u [domain\]user
[/p [password]]] [/ru [domain\]user]
[/rp password] [/tr tasktorun] [/st starttime] [/ri runintrrrva]
[/et endtime | /du duration] [/k] [/sd startdate] [/ed enddate]
[enable | disable] [/it] [/z]
```

SCHTASKS/CREATE

创建计划任务。

```
schtasks /create [/s system [/u [domain\]user [/p [password]]]
[/ru [domain\]username [rp password]] /tn taskname /tr tasktorun
/sc schedulertype [/mo modifier] [/d day] [/i idletime]
[/st starttime] [/m month [, month ...]] [/sd startdate]
[/ed enddate] [/ri runintrrrva] [/et endtime | /du duration] [/k]
[/it | /np] [/z] [/f] [/xml xmlfile]
```

SCHTASKS/DELETE

移除不再需要运行的计划任务。

```
schtasks /delete /tn {TaskName | *} [/f] [/s computer
[/u [domain\]user [/p [password]]]
```

SCHTASKS/END

终止运行中的任务。

```
schtasks /end /tn taskname [/s computer [/u [domain\]user  
[/p [password]]]]
```

SCHTASKS/QUERY

显示本地计算机或指定计算机上的计划任务。

```
schtasks /query [/s computer [/u [domain\]user [/p [password]]]]  
[/fo {table | list | csv} | /xml] [/nh] [/v] [/tn {TaskName}]
```

SCHTASKS/RUN

启动一个计划任务。

```
schtasks /run /tn taskname [/s computer [/u [domain\]user  
[/p [password]]]]
```

SERVERMANAGERCMD

安装或移除角色、角色服务与功能，也可以用于列出已安装的角色、角色服务与功能。
用于查询的语法格式如下：

```
servermanagercmd -query [queryfile.xml] [-logPath logfile.txt]  
servermanagercmd -version
```

用于安装的语法格式如下：

```
servermanagercmd -install component [-resultPath resultfile.xml]  
[-restart] | [-whatif]] [-logPath logfile.txt] [-allSubFeatures]
```

用于移除的语法格式如下：

```
servermanagercmd -remove component [-resultPath resultfile.xml]  
[-restart] | [-whatif]] [-logPath logfile.txt]
```

用于使用answer文件安装或移除的语法格式如下：

```
servermanagercmd -inputPath answerfile.xml  
[-resultPath resultfile.xml]  
[-restart] | [-whatif]] [-logPath logfile.txt]
```

SET

显示或修改Windows环境变量，也可以用于在命令行中评估数字表达式。

```
set [variable=[string]]  
set /a expression  
set /p variable=[promptstring]
```


SET-SERVICE

Windows PowerShell命令，用于修改系统服务的配置。

```
set-service [-name] servicename [-displayname displayname]
[-description description]
[-startuptype {Automatic|Manual|Disabled}] [-whatif] [-config]
[parameters]
```

SETLOCAL

在批处理文件中开始变量的局部化。

```
setlocal
setlocal {enableext | disableext}
```

SFC

扫描并验证受保护的系统文件。

```
sfc [/scannow] [/verifyonly] [/scanfile=file] [/verifyfile=file]
[/offwindir=offlinewindowsdirectory /offbootdir=offlinebootdirectory]
```

SHIFT

移动脚本中可替换变量的位置。

```
shift [/n]
```

SHUTDOWN

关闭或重启计算机。

```
shutdown [/i | /l | /s | /r | /g | /a | /p | /h | /e] [/f]
[/m \\computerName] [/t nn] [/d [p|u:]n1:n2 [/c "comment"]]
```

SORT

对输入进行排序。

```
[command |] sort [/r] [/+n] [/m kb] [/l locale] [/rec recordbytes]
[drive1:][path1]filename1 [/t [drive2:][path2]]
[/o [drive3:][path3]filename3]
```

注解 在本实例中，| 是管道符号。

START

启动一个新的命令shell窗口，用来运行指定的程序或命令。

```
start ["title"] [/d path] [/i] [/min] [/max] [/separate | /shared]
[/wait] [/b] [/low | /belownormal | /normal | /abovenormal]
| /high | /realtime] [/affinity hh] [command/program] [parameters]
```

STOP-PROCESS

Windows PowerShell命令，用于终止一个或多个运行中的进程。

```
stop-process -id [id1,id2,...] [-confirm] [-passthru] [-whatif]
[parameters]
stop-process -inputobject processname1, processname2,... [-passthru]
[-whatif] [-config] [parameters]
stop-process -name processname1, processname2,... [-confirm]
[-passthru] [-whatif] [parameters]
```

STOP-SERVICE

Windows PowerShell命令，用于终止一个或多个运行中的服务。

```
stop-service [-displayname [servicename1, servicename2,...]]
-include [servicename1, servicename2,...]
-exclude [servicename1, servicename2,...]
stop-service [-name] [servicename1, servicename2,...]
-include [servicename1, servicename2,...]
-exclude [servicename1, servicename2,...]
```

注解 Windows PowerShell还包括用于启动（start-service）、重启（restart-service）、挂起（suspend-service）、恢复（resume-service）服务的命令，其语法格式与stop-service一样。

SUBST

将路径映射到驱动器盘符。

```
subst [drive1: [drive2:]path]
subst drive1: /d
```

SYSTEMINFO

显示详尽的配置信息。

```
systeminfo [/s computer [/u [domain\]user [/p [pwd]]]]
[/fo {table|list|csv}] [/nh]
```

TAKEOWN

允许管理员拥有一个或多个文件的主导权。

```
takeown [/s computer [/u [domain\]user [/p [pwd]]]] /f filename
[/a] [/r [/d prompt]]
```

TASKKILL

根据进程名或进程ID终止运行中的进程。

```
taskkill [/s computer] [/u [domain\]user [/p pwd]] {[/fi filter1  
[/fi filter2 [ ... ]]} [/pid ID|/im imgName] [/f][/t]
```

TASKLIST

根据进程名或进程ID列出所有运行中的进程。

```
tasklist [/s computer [/u [domain\]user [/p [password]]]]  
[{/m module | /svc | /v}] [/fo {table | list | csv}] [/nh]  
[/fi filtername [/fi filtername2 [ ... ]]]
```

TIME

显示或设置系统时间。

```
time [time | /T]
```

TIMEOUT

在批脚本中设置超时周期或等待按键。

```
TIMEOUT /t timeout [/nobreak]
```

TITLE

为命令shell窗口设置标题。

```
title [string]
```

TRACERPT

从踪迹日志中生成踪迹报告。

```
tracertpt {[-l] logfile1 logfile2 ... | [-o outputfile] |  
-rt sessionname1 sessionname2 ...} [-of <CSV | EVTX | XML>]  
[-lr] [-summary summaryreportfile] [-report reportfilename]  
[-f <XML | HTML>] [-df schemafile] [-int dumpeventfile] [-rts]  
[-tmf tracedefinitionfile] [-tp tracefilepath] [-gmt] [-i imagepath]  
[-pdb symbolpath] [-r1 n] [-export schemaexportfile]  
[-config configfile] [-y]
```

TRACERT

显示计算机之间的联通路径。

```
tracert [-d] [-h maximumhops] [-j hostlist] [-w timeout]
[-r] [-s sourceaddress] [-4] [-6] targetname
```

TYPE

显示文本文件的内容。

```
type [drive:][path]filename
```

TYPEPERF

显示或记录指定计数器的性能数据。

```
typeperf [{counter1 counter2 ...} | -cf counterfile] [-sc numsamples]
[-si {[[hh:]mm:]ss} [-o logfile] [-f {CSV | TSV | BIN | SQL}]]
[-s server] [-y]
typeperf {-q object | -qx object} [-sc numsamples ]
[-si {[[hh:]mm:]ss} [-o logfile] [-f {CSV | TSV | BIN | SQL}]]
[-s server] [-y]
```

VER

显示Windows版本。

```
ver
```

VERIFY

让Windows验证文件是否正确写入到磁盘。

```
verify [on | off]
```

VOL

显示磁盘卷标与序列号。

```
vol [drive:]
```

WAITFOR

规定计算机等到特定信号后再继续执行。

发送信号的语法格式如下：

```
waitfor [/s computer [/u [domain\]user [/p [pwd]]]] /si signal
```

等待信号的语法格式如下：

```
waitfor [/t timeout] signal
```

WBADMIN

执行或计划备份与恢复操作，该命令仅适用于Windows Server 2008、Windows Vista Business、Enterprise、Ultimate版。

用于激活备份的语法格式如下：

```
wbadmin enable backup  
[-addtarget:{BackupTargetDisk}]  
[-removetarget:{BackupTargetDisk}]  
[-schedule:TimeToRunBackup] [-include:VolumesToInclude]  
[-allCritical] [-quiet]
```

用于禁用备份的语法格式如下：

```
wbadmin disable backup [-quiet]
```

用于启动备份的语法格式如下：

```
wbadmin start backup  
[-backupTarget:{TargetVolume | TargetNetworkShare}]  
[-include:VolumesToInclude] [-allCritical]  
[-noVerify] [-user:UserName] [-password:Password]  
[-noInheritAcl] [-vssFull] [-quiet]
```

用于终止当前备份任务的语法格式如下：

```
wbadmin stop job [-quiet]
```

用于列出可用磁盘的语法格式如下：

```
wbadmin get disks
```

用于获取当前备份任务状态的语法格式如下：

```
wbadmin get status
```

用于获取可用备份列表的语法格式如下：

```
wbadmin get versions  
[-backupTarget:{VolumeName | NetworkSharePath}]  
[-machine:BackupMachineName]
```

用于启动系统状态备份的语法格式如下：

```
wbadmin start systemstatebackup  
-backupTarget:{VolumeName} [-quiet]
```

用于启动系统状态恢复的语法格式如下：

```
wbadmin start systemstaterecovery  
-version:VersionIdentifier -showsummary  
[-backupTarget:{VolumeName | NetworkSharePath}]  
[-machine:BackupMachineName]
```

```
[-recoveryTarget:TargetPathForRecovery]  
[-authsysvol] [-quiet]
```

用于删除系统状态备份的语法格式如下:

```
wbadmin delete systemstatebackup  
-keepVersions: NumberCopiesToKeep | -version VersionID |  
-deleteOldest  
[-backupTarget:{VolumeName}] [-machine:BackupMachineName]  
[-quiet]
```

WHERE

显示匹配搜索模式的文件列表。

```
where [/r dir] [/q] [/f] [/t] pattern  
where [/q] [/f] [/t] $env:pattern  
where [/q] [/f] [/t] path:pattern
```

WHOAMI

显示当前用户的登录与安全信息。

```
whoami [/upn | /fqdn | /logonid]  
whoami {[/user] [/groups] [/priv]} [/fo {table|list|csv}] [/nh]  
whoami /all [/fo {table|list|csv}] [/nh]
```



如第17章所讲述的，网络服务shell（Netsh）是一个命令行脚本工具，可用于在本地计算机与远程计算机上对多种网络服务配置进行管理。Netsh提供了单独的命令提示符，可以在交互模式或非交互模式下使用。在Netsh命令环境下，可以操作复杂的上下文与命令。

本附录提供了Netsh的简明参考，有助于读者在不同的上下文中浏览并快速发现需要使用的命令。上下文是以字母顺序列出的，每个上下文中的子上下文与命令也是以字母顺序列出的。要记住的是，有些上下文只有在安装了相关的角色、角色服务与功能后才可用或才能正确起作用。

Netsh

Netsh上下文是网络服务shell中的顶级上下文，表B-1总结了这一级中的可用命令。不管操作的哪一级上下文，都可以使用命令..返回到上一级上下文。如果当前级内某子上下文是可用的，则可以键入该子上下文名，来访问该子上下文及其命令。因而，如果当前操作的是顶级的Netsh上下文，现在需要切换到Advfirewall上下文，可以键入**advfirewall**。如果后来需要返回到顶级的Netsh上下文，可以键入命令..。

表B-1 Netsh上下文中的可用命令

命 令	描 述
..	返回到上一级上下文
abort	摒弃脱机模式下所作的改变
add helper	安装一个帮助者DLL
advfirewall	切换到netsh advfirewall上下文
alias	添加一个别名
bridge	切换到netsh bridge上下文
bye	退出程序
commit	接受脱机模式下所作的改变
delete helper	移除一个帮助者DLL
dhcp	如果已经安装了DHCP Server角色，则切换到netsh dhcp上下文，否则切换到netsh dhcpclient上下文
dhcpclient	切换到netsh dhcpclient上下文
dump	显示上下文内的设置配置脚本
exec	运行脚本文件
exit	退出程序

(续)

命 令	描 述
firewall	切换到netsh firewall上下文
http	切换到netsh http上下文
interface	切换到netsh interface上下文
ipsec	切换到netsh ipsec上下文
lan	切换到netsh lan上下文
nap	切换到netsh nap上下文
netio	切换到netsh netio上下文
nps	切换到netsh nps上下文
offline	将当前模式设置为脱机
online	将当前模式设置为联机
p2p	切换到netsh p2p上下文
popd	从栈中弹出上下文
pushd	将当前上下文压入栈
quit	退出程序
ras	切换到netsh ras上下文
routing	切换到netsh routing上下文
rpc	切换到netsh rpc上下文
set file	将控制台输出复制到文件
set machine	设置当前的操作主机
set mode	将当前模式设置为联机或脱机
show alias	列出所有已定义的别名
show helper	列出所有顶级帮助者
show mode	显示当前模式
unalias	删除一个别名
winhttp	切换到netsh winhttp上下文
winsock	切换到netsh winsock上下文
wlan	切换到netsh wlan上下文

Netsh Advfirewall

Netsh Advfirewall上下文可以用于查看与管理高级安全Windows防火墙的设置，表B-2总结了该上下文及其相关子上下文中可用的命令。

表B-2 Netsh Advfirewall的命令与子上下文

上下文/命令	描 述
netsh advfirewall	
consec	切换到netsh advfirewall consec上下文
dump	显示上下文中设置配置脚本
export	将当前策略导出到文件
firewall	切换到netsh advfirewall firewall上下文
import	将策略文件导入到当前策略库

(续)

上下文/命令	描 述
monitor	切换到netsh advfirewall monitor上下文
reset	将策略重置为默认的out-of-box策略
set allprofiles	在所有配置文件中设置属性
set currentprofile	在活跃配置文件中设置属性
set domainprofile	在域配置文件中设置属性
set global	设置全局属性
set privateprofile	在私有配置文件中设置属性
set publicprofile	在公开配置文件中设置属性
set store	为当前交互式会话设置策略库
show allprofiles	显示所有配置文件的属性
show currentprofile	显示活跃配置文件的属性
show domainprofile	显示域配置文件的属性
show global	显示全局属性
show privateprofile	显示私有配置文件的属性
show publicprofile	显示公开配置文件的属性
show store	显示当前交互式会话的策略库
netsh advfirewall consec	
add rule	添加一条新的连接安全规则
delete rule	删除所有匹配的连接安全规则
dump	显示上下文中设置配置脚本
set rule	为现存规则设置新的属性值
show rule	显示指定的连接安全规则
netsh advfirewall firewall	
add rule	添加一条新的入站/出站防火墙规则
delete rule	删除所有匹配的入站/出站规则
dump	显示上下文中设置配置脚本
set rule	为现存规则设置新的属性值
show rule	显示指定的防火墙规则
netsh advfirewall monitor	
delete	删除所有匹配的安全关联
dump	显示上下文中设置配置脚本
show	显示所有匹配的安全关联

Netsh Bridge

Netsh Bridge上下文中可以查看与管理网桥的设置，表B-3总结了该上下文中可用的命令。

表B-3 Netsh Bridge上下文中的命令

命 令	描 述
dump	显示上下文中设置配置脚本

(续)

命 令	描 述
install	安装与当前上下文对应的组件
set adapter	修改指定适配器的网桥配置
show adapter	显示配置为单一网桥的适配器
uninstall	移除与当前上下文对应的组件

Netsh Dhcp

Netsh Dhcp上下文与相关的子上下文可用于管理DHCP服务器的配置。要注意的是, 这些上下文只有在DHCP Server角色服务已经在Windows服务器中安装后才是可用的。如果DHCP Server角色服务尚未安装, 则使用Dhcp命令会打开Dhcpclient上下文。

Netsh Dhcp上下文中支持如下的命令。

- **server[\\ServerName or \\IPAddress]**。将上下文切换到指定的服务器。
- **add server**。向目录服务中的授权服务器列表添加一台DHCP服务器。
- **delete server**。从目录服务中的授权服务器列表删除一台DHCP服务器。
- **show server**。显示当前域目录服务中的所有DHCP服务器。

表B-4、表B-5、表B-6分别总结了Netsh Dhcp Server上下文、Netsh Dhcp Server V4上下文、Netsh Dhcp Server V6上下文中可用的命令。

表B-4 Netsh Dhcp Server上下文中的命令

命 令	描 述
add class	向服务器中添加一个类
add mscope	向服务器中添加多播范围
add optiondef	向服务器中添加新选项
add scope	向服务器中添加范围
backup	备份配置
delete class	从服务器中删除特定类
delete dnscredentials	删除DNS动态更新要使用的凭据
delete mscope	从服务器中删除多播范围
delete optiondef	从服务器中删除选项
delete optionvalue	从服务器中删除选项值
delete superscope	从服务器中删除superscope
dump	将配置信息转储到文本文件
export	将配置信息导出到文件
import	从文件中导入配置信息
initiate auth	发起到服务器的授权尝试
initiate reconcile	检测与协调服务器下的所有范围的数据库
mscope<mscope-name>	切换到由Mscope名标识的mscope
restore	恢复配置
scope<scope-ip-address>	切换到由IP地址标识的范围
set auditlog	设置服务器的审计日志参数

(续)

命 令	描 述
set bindings	设置服务器的接口绑定
set databasebackupinterval	设置当前服务器的数据库备份时间间隔
set databasebackuppath	设置当前服务器的数据库备份路径
set databasecleanupinterval	设置数据库清空时间间隔
set databaseloggingflag	设置/重置数据库登录标记
set databasename	设置服务器数据库文件名
set databasepath	设置服务器数据库文件路径
set databaserestoreflag	设置/重置数据库恢复标记
set detectconflictretry	设置DHCP服务器的冲突检测尝试次数
set dnsconfig	设置服务器的DNS动态更新配置
set dnscredentials	设置DNS动态更新使用的凭据
set napdeffail	设置服务器的NAP默认失败状态
set napstate	设置服务器的NAP状态
set optionvalue	设置服务器的全局选项值
set server	设置服务器模式下的当前服务器
set userclass	设置随后操作的全局用户类名
set vendorclass	设置随后操作的全局销售商类名
show all	显示服务器的所有信息
show auditlog	显示服务器的审计日志参数
show bindings	显示服务器的接口绑定
show class	显示服务器的所有可用类
show dbproperties	显示服务器数据库配置信息
show detectionconflictretry	显示冲突检测重试设置
show dnsconfig	显示服务器的DNS动态更新配置
show dnscredentials	显示当前设置的DNS凭据
show mibinfo	显示服务器的MIBInfo
show mscope	显示服务器的多播范围
show napdeffail	显示服务器的NAP默认失败状态
show napstate	显示服务器的NAP状态
show optiondef	显示服务器的所有选项
show optionvalue	显示服务器设置的所有选项值
show scope	显示服务器下所有可用的范围
show server	显示当前服务器
show serverstatus	显示服务器的当前状态
show superscope	显示服务器下所有可用的超级范围
show userclass	显示当前设置的用户类名
show vendorclass	显示当前设置的销售商类名
show version	显示服务器的当前版本
V4	切换到netsh dhcp server v4上下文
V6	切换到netsh dhcp server v6上下文

表B-5 Netsh Dhcp Server V4上下文的命令

命 令	描 述
add class	向服务器中添加一个类
add mscope	向服务器中添加多播范围
add optiondef	向服务器中添加新选项
add scope	向服务器中添加范围
delete class	从服务器中删除特定类
delete dnscredentials	删除DNS动态更新要使用的凭据
delete mscope	从服务器中删除多播范围
delete optiondef	从服务器中删除选项
delete optionvalue	从服务器中删除选项值
delete scope	从服务器中删除范围
delete superscope	从服务器中删除超级范围
dump	将配置信息转储到文本文件
export	将配置信息导出到文件
import	从文件中导入配置信息
set bindings	设置服务器的接口绑定
set detectconflictretry	设置DHCP服务器的冲突检测尝试次数
set dnsconfig	设置服务器的DNS动态更新配置
set dnscredentials	设置DNS动态更新使用的凭据
set napdeffail	设置服务器的NAP默认失败
set napstate	设置服务器的NAP默认失败状态
set optionvalue	设置服务器的全局选项值
set userclass	设置随后操作的全局用户类名
set vendorclass	设置随后操作的全局销售商类名
show all	显示服务器的所有信息
show bindings	显示服务器的接口绑定
show class	显示服务器的所有可用类
show detectionconflictretry	显示冲突检测重试设置
show dnsconfig	显示服务器的DNS动态更新配置
show dnscredentials	显示当前设置的DNS凭据
show mibinfo	显示服务器的MIBInfo
show mscope	显示服务器的多播范围
show napdeffail	显示服务器的NAP默认失败状态
show napstate	显示服务器的NAP状态
show optiondef	显示服务器的所有选项
show optionvalue	显示服务器设置的所有选项值
show scope	显示服务器下所有可用的范围
show superscope	显示服务器下所有可用的超级范围
show userclass	显示当前设置的用户类名
show vendorclass	显示当前设置的销售商类名

表B-6 Netsh Dhcp Server V6上下文的命令

命 令	描 述
add class	向服务器中添加一个类
add optiondef	向服务器中添加DHCPv6选项
add scope	向服务器中添加范围
delete class	从服务器中删除特定类
delete optiondef	从服务器中删除选项
delete optionvalue	从服务器中删除选项值
delete scope	从服务器中删除范围
dump	将配置信息转储到文本文件
export	将配置信息导出到文件
import	从文件中导入配置信息
set bindings	设置服务器的接口绑定
set dnsconfig	设置服务器的DNS动态更新配置
set preferredlifetime	设置DHCP服务器上已发布lease的首选生存时间
set rapidcommitflag	设置服务器的全局快速commit标记
set t1	设置DHCP服务器上已发布lease的T1值
set t2	设置DHCP服务器上已发布lease的T2值
set unicastflag	设置服务器的全局单播标记
set userclass	设置随后操作的全局用户类名
set validlifetime	设置DHCP服务器上已发布lease的有效生存时间
set vendorclass	设置随后操作的全局销售商类名
show bindings	显示服务器的接口绑定
show class	显示服务器的所有可用类
show dnsconfig	显示服务器的DNS动态更新配置
show mibinfo	显示服务器的MIBInfo
show optiondef	显示服务器的所有DHCPv6选项
show optionvalue	显示服务器设置的所有选项值
show preferredlifetime	设置DHCP服务器上已发布lease的首选生存时间
show rapidcommitflag	设置服务器的全局快速commit标记
show scope	显示服务器下所有可用范围
show t1	显示已发布lease的T1值
show t2	显示已发布lease的T2值
show unicastflag	显示单播标记
show userclass	显示当前设置的用户类名
show validlifetime	显示有效生存时间
show vendorclass	显示当前设置的销售商类名

Netsh Dhcpclient

Netsh Dhcpclient上下文中可以激活或禁用对DHCP通信的追踪，该上下文支持如下一些命令。

- **trace enable**。激活对DHCP客户端与DHCP QEC的追踪。
- **trace disable**。禁用对DHCP客户端与DHCP QEC的追踪。

Netsh Firewall

Netsh Firewall上下文可用于对Windows防火墙进行管理，表B-7总结了这一上下文中所有可用命令。

表B-7 Netsh Firewall上下文中可用命令

命 令	描 述
add allowedprogram	添加防火墙允许通过的程序配置
add portopening	添加防火墙端口配置
delete allowedprogram	删除防火墙允许通过的程序配置
delete portopening	删除防火墙端口配置
dump	显示上下文中配置脚本设置
reset	将防火墙配置重置为默认设置
set allowedprogram	设置防火墙允许通过的程序配置
set icmpsetting	设置防火墙ICMP配置
set logging	设置防火墙记录配置
set multicastbroadcastresponse	设置防火墙多播/广播响应配置
set notifications	设置防火墙通知配置
set opmode	设置防火墙操作配置
set portopening	设置防火墙端口配置
set service	设置防火墙服务配置
show allowedprogram	设置防火墙允许通过的程序配置
show config	显示防火墙配置
show currentprofile	显示当前防火墙配置文件
show icmpsetting	显示防火墙ICMP配置
show logging	显示防火墙记录配置
show multicastbroadcastresponse	显示防火墙多播/广播响应配置
show notifications	显示防火墙通知配置
show opmode	显示防火墙操作配置
show portopening	显示防火墙端口配置
show service	显示防火墙服务配置
show state	显示当前防火墙状态

Netsh HTTP

Netsh Http上下文可用于管理HTTP监听程序的配置，表B-8总结了该上下文中可用的命令。

表B-8 Netsh Http上下文中可用命令

命 令	描 述
add iplisten	向IP监听列表中添加一个IP地址
add sslcert	为IP地址与端口添加SSL服务器证书绑定
add timeout	为服务添加全局超时
add urlacl	添加一个URL保留条目

(续)

命 令	描 述
delete cache	从HTTP服务内核URI缓存中删除条目
delete iplisten	从IP监听列表中删除一个IP地址
delete sslcert	从IP地址与端口删除SSL证书绑定
delete timeout	删除全局超时
delete urlacl	删除URL保留
dump	显示上下文内的配置脚本设置
flush logbuffer	清空日志文件的内部缓冲区
show cachestate	列出缓存的URI资源及其相关联的属性
show iplisten	显示IP监听列表中的所有IP地址
show servicestate	显示HTTP服务的快照
show sslcert	显示IP地址与端口的SSL证书绑定
show timeout	显示服务的超时值
show urlacl	显示URL命名空间保留

Netsh Interface

Netsh Interface上下文中可以配置计算机网络接口与IP地址，如表B-9中所显示的，该上下文还包括了很多相关的上下文。

表B-9 Netsh Interface上下文中的命令

命 令	描 述
6to4	切换到netsh interface 6to4上下文
add interface	为路由器添加接口
delete interface	从路由器删除接口
dump	显示上下文内的配置脚本设置
ipv4	切换到netsh interface ipv4上下文
ipv6	切换到netsh interface ipv6上下文
isatap	切换到netsh interface isatap上下文
portproxy	切换到netsh interface portproxy上下文
reset all	重置信息
set credentials	设置连接到接口使用的凭据
set interface	设置接口参数
show credentials	显示连接到接口使用的凭据
show interface	显示接口
tcp	切换到netsh interface tcp上下文
teredo	切换到netsh interface teredo上下文

表B-10、表B-11分别总结了配置Ipv4、Ipv6网络接口时可用的命令，表B-12总结了其他Netsh Interface子上下文中可用的命令。

表B-10 用于Netsh Interface Ipv4上下文的命令

命 令	描 述
add address	为指定的接口添加静态IP地址或默认网关
add dnsserver	添加静态DNS服务器地址
add neighbors	添加邻居地址
add route	为接口添加路由
add winsserver	添加静态WINS服务器地址
delete address	为指定的接口删除静态IP地址或默认网关
delete arpcache	清空某个或所有接口的ARP缓存
delete destinationcache	删除目标缓存
delete dnsserver	从指定的接口删除DNS服务器
delete neighbors	删除邻居缓存
delete route	删除路由
delete winsserver	从指定的接口删除WINS服务器
dump	显示上下文内的脚本配置设置
install	安装IP协议
reset	重置IP配置
set address	为接口设置IP地址或默认网关
set compartment	修改区域配置参数
set dnsserver	设置DNS服务器模式与地址
set dynamicportrange	修改动态端口分配使用的端口范围
set global	修改全局配置的常规参数
set interface	修改IP地址的接口配置参数
set neighbors	设置邻居地址
set route	修改路由参数
set subinterface	修改子接口配置参数
set winsserver	设置WINS服务器模式与地址
show addresses	显示IP地址配置
show compartments	显示区域参数
show config	显示IP地址与附加信息
show destinationcache	显示目的缓存条目
show dnsservers	显示DNS服务器地址
show dynamicportrange	显示动态端口范围配置参数
show global	显示全局配置参数
show icmpstats	显示ICMP统计信息
show interfaces	显示接口参数
show ipaddresses	显示当前IP地址
show ipnetto-media	显示IP net-to-media映射
show ipstats	显示IP统计信息
show joins	显示添加的多播组
show neighbors	显示邻居缓存条目

(续)

命 令	描 述
show offload	显示卸载信息
show route	显示路由表条目
show subinterfaces	显示子接口参数
show tcpconnections	显示TCP连接
show tcpstats	显示TCP统计信息
show udpconnections	显示UDP连接
show udpstats	显示UDP统计信息
show winsservers	显示WINS服务器地址
uninstall	卸载IP协议

表B-11 Netsh Interface Ipv6上下文中的命令

命 令	描 述
6to4	切换到netsh interface ipv6 6to4上下文
add address	为指定的接口添加静态IP地址或默认网关
add dnsserver	添加静态DNS服务器地址
add neighbors	添加邻居地址
add potentialrouter	将路由器添加到接口上潜在的路由器列表
add prefixpolicy	添加一个前缀策略条目
add route	为接口添加路由
add v6v4tunnel	创建Ipv6-in-Ipv4点到点隧道
delete address	从指定接口删除IP地址或默认网关
delete destinationcache	删除目标缓存
delete dnsserver	从指定的接口删除DNS服务器
delete neighbors	删除邻居缓存
delete potentialrouter	从接口上潜在路由器列表中删除路由器
delete prefixpolicy	删除前缀策略条目
delete route	删除路由
dump	显示上下文内的脚本配置设置
isatap	切换到netsh interface ipv6 isatap上下文
reset	重置IP配置
set address	为接口设置IP地址或默认网关
set compartment	修改区域配置参数
set dnsserver	设置DNS服务器模式与地址
set dynamicportrange	修改动态端口分配使用的端口范围
set global	修改全局配置的常规参数
set interface	修改IP地址的接口配置参数
set neighbors	设置邻居地址
set prefixpolicy	修改前缀策略信息

(续)

命 令	描 述
set privacy	修改隐私配置参数
set route	修改路由参数
set subinterface	修改子接口配置参数
set teredo	设置Teredo状态
show addresses	显示当前IP地址
show compartments	显示区域参数
show destinationcache	显示目的缓存条目
show dnsservers	显示DNS服务器地址
show dynamicportrange	显示动态端口范围配置参数
show global	显示全局配置参数
show interfaces	显示接口参数
show ipstats	显示IP统计信息
show joins	显示添加的多播组
show neighbors	显示邻居缓存条目
show offload	显示卸载信息
show potentialrouters	显示潜在的路由器
show prefixpolicies	显示前缀策略条目
show privacy	显示隐私配置参数
show route	显示路由表条目
show siteprefixes	显示站点prefix表条目
show subinterfaces	显示子接口参数
show tcpstats	显示TCP统计信息
show teredo	显示Teredo状态
show udpstats	显示UDP统计信息

表B-12 用于Netsh Interface其他子上下文的命令

上 下 文	描 述
netsh interface 6to4	
dump	显示上下文内的配置脚本设置
set interface	设置6to4接口配置信息
set relay	设置6to4转播信息
set routing	设置6to4路由信息
set state	设置6to4状态
show interface	显示6to4接口配置信息
show relay	显示6to4转播信息
show routing	显示6to4路由状态
state	显示6to4状态
netsh interface ipv6 6to4	
dump	显示上下文内的配置脚本设置

(续)

上 下 文	描 述
set interface	设置6to4接口配置信息
set relay	设置6to4转播信息
set routing	设置6to4路由信息
set state	设置6to4状态
show interface	显示6to4接口配置信息
show relay	显示6to4转播信息
show routing	显示6to4路由状态
show state	显示6to4状态
netsh interface ipv6 isatap	
dump	显示上下文内的配置脚本设置
set router	设置ISATAP路由器信息
set state	设置ISATAP状态
show router	显示ISATAP路由器信息
show state	显示ISATAP状态
netsh interface isatap	
dump	显示上下文内的配置脚本设置
set router	设置ISATAP路由器信息
set state	设置ISATAP状态
show router	显示ISATAP路由器信息
show state	显示ISATAP状态
netsh interface portproxy	
add v4tov4	添加一个条目, 用来监听通过Ipv4的Ipv4与代理连接
add v4tov6	添加一个条目, 用来监听通过Ipv6的Ipv4与代理连接
add v6tov4	添加一个条目, 用来监听通过Ipv4的Ipv6与代理连接
add v6tov6	添加一个条目, 用来监听通过Ipv6的Ipv6与代理连接
delete v4tov4	删除一个条目, 该条目用于监听通过Ipv4的Ipv4与代理连接
delete v4tov6	删除一个条目, 该条目用于监听通过Ipv6的Ipv4与代理连接
delete v6tov4	删除一个条目, 该条目用于监听通过Ipv4的Ipv6与代理连接
delete v6tov6	删除一个条目, 该条目用于监听通过Ipv6的Ipv6与代理连接
dump	显示上下文内的配置脚本设置
reset	重置portproxy配置状态
set v4tov4	更新一个条目, 用来监听通过Ipv4的Ipv4与代理连接
set v4tov6	更新一个条目, 用来监听通过Ipv6的Ipv4与代理连接
set v6tov4	更新一个条目, 用来监听通过Ipv4的Ipv6与代理连接
set v6tov6	更新一个条目, 用来监听通过Ipv6的Ipv6与代理连接
show all	显示所有端口代理参数
show v4tov4	显示一些参数, 这些参数用于代理Ipv4到其他Ipv4端口的连接
show v4tov6	显示一些参数, 这些参数用于代理Ipv4到Ipv6的连接
show v6tov4	显示一些参数, 这些参数用于代理Ipv6到Ipv4的连接
show v6tov6	显示一些参数, 这些参数用于代理Ipv4到其他Ipv4端口的连接

(续)

上 下 文	描 述
netsh interface tcp	
add chimneyapplication	向TCP Chimney offload表添加一个应用程序
add chimneyport	为指定的本地源端口与远程目的端口向TCP Chimney offload表添加一个应用程序
delete chimneyapplication	从卸载表中删除一个TCP Chimney端口条目
dump	显示上下文内配置脚本设置
reset	将所有TCP参数重置为其各自的默认值
set global	设置全局TCP参数
show chimneyapplications	显示TCP Chimney offload表中的应用程序
show chimneyports	显示TCP Chimney offload表中的端口
show global	显示全局TCP参数
netsh interface teredo	
dump	显示上下文内配置脚本设置
set state	设置Teredo状态
show state	显示Teredo状态

Netsh Ipsec

使用Netsh Ipsec上下文及其相关子上下文，可以对Internet协议安全（IPsec）进行配置，表B-13总结了本上下文及其相关所有子上下文中可用的命令。

表B-13 用于Netsh Ipsec、Netsh Ipsec Dynamic、Netsh Ipsec Static 上下文中的命令

上下文/命令	描 述
netsh ipsec	
dump	显示上下文内配置脚本设置
dynamic	切换到netsh ipsec dynamic上下文
static	切换到netsh ipsec static上下文
netsh ipsec dynamic	
add mmpolicy	向SPD添加一条主模式策略
add qmpolicy	向SPD添加一条快速模式策略
add rule	向SPD添加一条规则及其相关联的过滤器
delete all	从SPD中删除所有的策略、过滤器及操作
delete mmpolicy	从SPD中删除一条主模式策略
delete qmpolicy	从SPD中删除一条快速模式策略
delete rule	从SPD中删除一条规则及其相关联的过滤器
delete sa	删除一个安全关联
dump	显示上下文内配置脚本设置
set config	设置IPsec配置及引导时间行为
set mmpolicy	修改SPD中的一条主模式策略

(续)

上下文/命令	描 述
set qmpolicy	修改SPD中的一条快速模式策略
set rule	修改SPD中的一条规则及其相关联的过滤器
show all	显示SPD中的策略、过滤器、SA以及一些统计信息
show config	显示IPsec配置
show mmfilter	显示SPD中的主模式过滤器的详细资料
show mmpolicy	显示SPD中的主模式策略的详细资料
show mmsas	显示SPD中主模式安全关联
show qmfilter	显示SPD中的快速模式过滤器的详细资料
show qmpolicy	显示SPD中的快速模式策略的详细资料
show qmsas	显示SPD中快速模式安全关联
show rule	显示SPD中规则详细资料
show stats	显示SPD中IPsec与IKE统计信息
netsh ipsec static	
add filter	向过滤器列表中添加一个过滤器
add filteraction	创建一个过滤器操作
add filterlist	创建一个空的过滤器列表
add policy	创建一条策略，使用默认的响应规则
add rule	为指定策略创建一条规则
delete all	删除所有的策略、过滤器列表与过滤器操作
delete filter	从过滤器列表中删除一个过滤器
delete filteraction	删除一个过滤器操作
delete filterlist	删除一个过滤器列表
delete policy	删除一条策略及其规则
delete rule	从策略中删除一条规则
dump	显示上下文内配置脚本设置
exportpolicy	从策略库中导出所有策略
importpolicy	将策略从文件中导入到策略库
set batch	设置批量更新模式
set defaultrule	修改策略的默认响应规则
set filteraction	修改过滤器操作
set filterlist	修改过滤器列表
set policy	修改策略
set rule	修改规则
set store	设置当前策略库
show all	显示所有策略及其相关信息的详细资料
show filteraction	显示过滤器操作的详细资料
show filterlist	显示过滤器列表的详细资料
show gpoassignedpolicy	显示组指定策略的详细资料

(续)

上下文/命令	描 述
show policy	显示策略的详细资料
show rule	显示规则的详细资料
show store	显示当前策略库

Netsh Lan

在Wired AutoConfig服务处于激活状态时，可以通过Netsh Lan上下文查看与管理自动配置的有线网络接口及其相关的配置与设置。表B-14总结了这一上下文中可用的命令。

表B-14 Netsh Lan上下文内的命令

命 令	描 述
add profile	向计算机上指定网络接口上添加LAN配置文件
delete profile	从一个或多个网络接口上删除LAN配置文件
dump	显示上下文内配置脚本设置
export	将LAN配置文件保存为XML文件
reconnect	在一个网络接口上重新连接
set autoconfig	激活或禁用网络接口上的自动配置
set profileparameter	对无线配置中的认证与签名机制进行配置
set tracing	激活或禁用追踪
show interfaces	显示系统中当前有线网络接口列表
show profiles	显示计算机上当前配置的有线网络配置列表
show settings	显示有线LAN当前的全局设置
show tracing	显示有线LAN追踪处于激活还是禁用状态

Netsh Nap

Netsh Nap上下文可以用于对网络访问保护（NAP）的配置进行管理。表B-15总结了这一上下文及其相关的Client、Hra子上下文中可用的命令。

表B-15 Netsh Nap及其相关子上下文内的命令

命 令	描 述
netsh nap	
client	切换到netsh nap client上下文
dump	显示上下文内配置脚本设置
hra	切换到netsh nap hra上下文
reset configuration	重置NAP配置
show configuration	显示NAP配置
netsh nap client	
add server	添加可信服务器配置
add trustedservergroup	添加可信服务器组配置

(续)

命 令	描 述
delete server	删除可信服务器配置
delete trustedservergroup	删除可信服务器组配置
dump	显示上下文内配置脚本设置
export	导出配置设置
import	导入配置设置
rename server	对可信服务器组内现存可信服务器的URL进行重命名
rename trustedservergroup	对可信服务器组进行重命名
reset configuration	重置NAP客户端配置
reset csp	重置CSP配置
reset enforcement	重置强制配置
reset hash	重置hash配置
reset server	重置可信服务器配置
reset tracing	重置追踪配置
reset trustedservergroup	重置可信服务器组配置
reset userinterface	重置用户接口配置
set csp	设置CSP配置
set enforcement	设置强制配置
set hash	设置hash配置
set server	设置可信服务器配置
set tracing	设置追踪配置
set userinterface	设置用户接口配置
show configuration	显示配置
show csps	显示CSP配置
show grouppolicy	显示组策略配置
show hashes	显示hash配置
show state	显示状态
show trustedservergroup	显示所有可信服务器组
netsh nap hra	
add asymmetrickey	添加非同步密钥配置
add caserver	添加CA服务器配置
add csp	添加CSP配置
add hash	添加hash配置
add useragent	添加用户agent配置
delete asymmetrickey	删除非同步密钥配置
delete caserver	删除CA服务器配置
delete csp	删除CSP配置
delete hash	删除hash配置
delete useragent	删除用户agent
dump	显示上下文内配置脚本设置

(续)

命 令	描 述
export	导出配置设置
import	导入配置设置
rename caserver	对CA服务器配置进行重命名
reset asymmetrickey	重置非同步密钥配置
reset caserver	重置CA服务器配置
reset configuration	重置配置
reset csp	重置CSP配置
reset hash	重置hash配置
reset opmode	重置HRA当前模式
reset templates	重置HRA证书模板配置
reset timeout	重置超时设置
reset userpolicyOIDs	重置策略OID配置
reset useragent	重置用户agent配置
reset validityperiod	重置HRA有效性周期
set caserver	设置CA服务器配置
set opmode	设置HRA模式
set templates	设置HRA证书模板配置
set timeout	设置超时配置
set userpolicyOIDs	设置策略OID配置
set validityperiod	设置证书的有效期（以分钟计数）
show asymmetrickeys	显示非同步密钥
show configuration	显示配置
show csps	显示CSP
show hashes	显示hash

Netsh Netio

Netsh Netio上下文可以用于添加、删除或列出网络绑定的过滤器，可用的命令包括下面列举的。

- ❑ **add bindingfilter**。添加绑定的过滤器。
- ❑ **delete bindingfilter**。删除绑定的过滤器。
- ❑ **dump**。显示配置脚本。
- ❑ **show bindingfilters**。显示所有绑定的过滤器。

Netsh Nps

Netsh Nps上下文可以用于管理网络策略服务器（NPS）的配置。在Windows Server 2008中，NPS取代了Windows Server 2008中的Internet认证服务（IAS）。表B-16总结了该上下文中可用的命令。

表B-16 Netsh Nps上下文中的命令

命 令	描 述
add client	添加客户端配置

(续)

命 令	描 述
add crp	添加连接请求策略配置
add np	添加网络策略配置
add registeredserver	在活动目录中注册NPS服务器
add remediationserver	添加矫正服务器配置
add remediationservergroup	添加矫正服务器组配置
add remoteserver	添加远程服务器配置
add remoteservergroup	添加远程服务器组配置
add shvtemplate	添加健康策略配置
delete client	删除客户端配置
delete crp	删除连接请求策略配置
delete np	删除网络策略配置
delete registeredserver	在活动目录中取消注册NPS服务器
delete remediationserver	删除矫正服务器配置
delete remediationservergroup	删除矫正服务器组配置
delete remoteserver	删除远程服务器配置
delete remoteservergroup	删除远程服务器组配置
delete shvtemplate	删除健康策略配置
dump	显示上下文内配置脚本设置
export	导出配置
import	导入配置
rename client	重命名客户端配置
rename crp	重命名连接请求策略配置
rename np	重命名网络策略配置
rename remediationserver	重命名矫正服务器配置
rename remediationservergroup	重命名矫正服务器组配置
rename remoteserver	重命名远程服务器配置
rename remoteservergroup	重命名远程服务器组配置
rename shvtemplate	重命名健康策略配置
reset client	重置客户端配置
reset config	重置配置
reset crp	重置连接请求策略配置
reset eventlog	重置事件日志配置
reset filelog	重置文件日志配置
reset np	重置网络策略配置
reset ports	重置端口配置
reset remediationserver	重置矫正服务器配置
reset remediationservergroup	重置矫正服务器组配置
reset remoteserver	重置远程服务器配置
reset remoteservergroup	重置远程服务器组配置

(续)

命 令	描 述
reset shv	重置系统健康验证器配置
reset shvtemplate	重置健康策略配置
reset sqllog	重置SQL日志配置
set client	设置客户端配置
set crp	设置连接请求策略配置
set eventlog	设置事件日志配置
set filelog	设置文件日志配置
set np	设置网络策略配置
set ports	设置端口配置
set remediationserver	设置矫正服务器配置
set remoteserver	设置远程服务器配置
set shv	设置系统健康验证器配置
set shvtemplate	设置健康策略配置
set sqllog	设置SQL日志配置
show client	显示客户端配置
show config	显示配置
show crp	显示连接请求策略配置
show crpconditionattributes	显示所有可用的连接请求策略条件属性
show crpprofileattributes	显示所有可用的连接请求策略配置属性
show eventlog	显示事件日志配置
show filelog	显示文件日志配置
show napserverinfo	显示NAP服务器信息
show np	显示网络策略配置
show npconditionattributes	显示所有可用的网络策略条件属性
show npprofileattributes	显示所有可用的网络策略配置属性
show ports	显示端口配置
show registeredserver	显示活动目录中NPS服务器的注册
show remediationserver	显示矫正服务器配置
show remediationservergroup	显示矫正服务器组配置
show remoteserver	显示远程服务器配置
show remoteservergroup	显示远程服务器组配置
show shv	显示系统健康验证器配置
show shvtemplate	显示健康策略配置
show sqllog	显示SQL日志配置
show vendors	显示所有可用的销售商

Netsh P2p

Netsh P2p上下文及其相关子上下文可以用于管理点对点网络（P2P）的配置，表B-16总结了该上

下文及其相关子上下文中可用的命令。

表B-17 Netsh P2p上下文中及其相关子上下文中的命令

上下文/命令	描 述
netsh p2p	
collab	切换到netsh p2p collab上下文
dump	显示上下文内配置脚本设置
group	切换到netsh p2p group上下文
idmgr	切换到netsh p2p idmgr上下文
pnrp	切换到netsh p2p pnrp上下文
netsh p2p collab	
contact	切换到netsh p2p collab contact上下文
dump	显示上下文内配置脚本设置
netsh p2p collab contact	
delete	从联系人数据库中删除一个联系人
dump	显示上下文内配置脚本设置
export	将联系人Me（代表当前用户）导出到一个文件
import	将联系人导入到联系人数据库
set	改变联系人数据
show contacts	显示联系人数据
show xml	显示联系人XML文件内容
netsh p2p group	
database	切换到netsh p2p group database上下文
dump	显示上下文内配置脚本设置
gping	检测到远程组端口的连通性
resolve	解析组内的一个参与方并列出其地址
show acl	列出访问控制列表（ACL）信息
show address	解析当前节点内的一个参与方并列出其地址
show statistics	列出给定<identity P2PID> <group P2PID>的数据库统计信息
netsh p2p idmgr	
delete group	从实体中删除组
delete identity	删除实体
dump	显示上下文内配置脚本设置
show groups	显示实体与组列表
show identities	显示实体列表
show statistics	显示实体统计信息
netsh p2p pnrp	
cloud	切换到netsh p2p pnrp cloud上下文
diagnostics	切换到netsh p2p pnrp diagnostics上下文
dump	显示上下文内配置脚本设置
peer	切换到netsh p2p pnrp peer上下文

(续)

上下文/命令	描 述
netsh p2p pnrp cloud	
dump	显示上下文内配置脚本设置
flush	清空缓存条目
repair	开始分割检测与恢复
set pnrpmode	修改PNRP模式配置参数
set seed	修改PNRP SeedServer模式配置参数
show initialization	显示cloud自举配置/状态
show list	显示cloud列表
show names	显示本地注册名
show pnrpmode	显示PNRP模式配置参数
show seed	显示PNRP SeedServer模式配置参数
show statistics	显示cloud统计信息
start	启动P2P网络cloud
synchronize host	将cloud与指定的主机同步
synchronize seed	将cloud与其种子服务器同步
netsh p2p pnrp diagnostics	
dump	显示上下文内配置脚本设置
ping	对PNRP节点进行ping操作
netsh p2p pnrp peer	
add registration	注册一个端点名
delete registration	取消端点名的注册
dump	显示上下文内配置脚本设置
enumerate	枚举指定cloud内的端点名
resolve	解析端点名
set machinename	设置PnrpAutoReg服务的配置信息
show convertedname	端点名到DNS名的互相转换
show machinename	显示PNRP机器名发布服务的配置信息
show registration	列出注册的端点名
tracert	解析带路径追踪的端点名

Netsh Ras

Netsh Ras上下文可以用于管理远程访问服务（RAS）、路由与远程访问服务（RRAS）的配置，表B-18总结了该上下文中可用的命令。

表B-18 Netsh Ras上下文中的命令

命 令	描 述
aaaa	切换到netsh ras aaaa上下文
add authtype	添加远程访问服务器协商使用的认证类型

(续)

命 令	描 述
add link	添加PPP协商使用的链路属性列表
add multilink	添加PPP协商使用的多链路类型列表
add registeredserver	将给定的Windows计算机注册为给定域内活动目录中远程访问服务器
delete authtype	删除远程访问服务器的一种认证类型
delete link	从PPP协商使用的链路属性列表进行删除
delete multilink	从PPP协商使用的多链路类型列表中进行删除
delete registeredserver	将给定的Windows计算机取消注册为给定域内活动目录中远程访问服务器
demanddial	切换到netsh ras demanddial上下文
diagnostics	切换到netsh ras diagnostics上下文
dump	显示上下文内配置脚本设置
ip	切换到netsh ras ip上下文
ipv6	切换到netsh ras ipv6上下文
set authmode	设置认证模式
set client	重置统计信息, 或断开远程客户端的连接
set conf	设置服务器的配置状态
set portstatus	重置RAS端口的统计信息
set type	设置计算机的路由器与RAS功能
set user	设置用户的远程访问属性
show activeservers	通过列出远程访问服务器广告显示活跃的RAS服务器
show authmode	显示认证模式
show authtype	显示当前激活的认证类型
show client	显示连接到计算机的远程访问客户端及其状态
show conf	显示服务器的配置状态
show link	显示PPP协商使用的链路属性
show multilink	显示PPP协商使用的多链路类型
show portstatus	显示RAS端口的当前状态
show registeredserver	显示某计算机是否已注册为给定域内的活动目录中远程访问服务器
show status	显示路由与远程访问服务器的状态
show type	显示计算机的路由器与RAS功能
show user	显示用户的远程访问属性

表B-19、表B-20、表B-21分别总结了Netsh Ras Aaaa上下文、Netsh Ras Demanddial上下文、Netsh Ras Diagnostics上下文中可用的命令。另外两个上下文Netsh Ras Ip与Netsh Ras Ipv6中可用的命令则在表B-22中总结。

表B-19 Netsh Ras Aaaa上下文中的命令

命 令	描 述
add acctserver	添加一个RADIUS记账服务器
add authserver	添加一个RADIUS认证服务器
delete acctserver	删除一个RADIUS记账服务器

(续)

命 令	描 述
delete authserver	删除一个RADIUS服务器
dump	显示上下文内配置脚本设置
set accounting	设置记账提供者
set acctserver	设置记账服务器的属性
set authentication	设置认证提供者
set authserver	设置认证服务器属性
set ipsecpolicy	设置L2TP连接的IPsec策略
show accounting	显示当前的记账提供者
show acctserver	显示用于记账的RADIUS服务器
show authentication	显示当前的认证提供者
show authserver	显示用于认证的RADIUS服务器
show ipsecpolicy	显示L2TP连接的IPsec策略

表B-20 Netsh Ras Demanddial上下文中的命令

命 令	描 述
add interface	添加一个新的请求拨号接口
delete interface	删除一个请求拨号接口
dump	显示上下文内配置脚本设置
set callbackdevice	设置用于请求拨号接口的回拨设备的回拨号
set credentials	设置请求拨号接口的拨号凭据
set interface	设置请求拨号接口的选项设置
set ppp	设置请求拨号接口的PPP选项
set security	设置请求拨号接口的安全选项
show callbackdevice	显示用于请求拨号接口的回拨设备的回拨号
show interface	显示请求拨号接口的设置
show ppp	显示请求拨号接口的PPP选项
show security	显示请求拨号接口的安全选项

表B-21 Netsh Ras Diagnostics上下文中的命令

命 令	描 述
dump	显示上下文内配置脚本设置
set cmtracing	激活/禁用连接管理者记录
set loglevel	设置RRAS的全局日志级别
set modemtracing	激活/禁用网络连接中调制解调器设置与消息追踪
set rastracing	激活/禁用对某组件的扩展追踪
set securityeventlog	激活或禁用安全事件记录。要查看安全事件日志, 可以使用事件查看器
set tracefacilities	激活/禁用对所有组件的扩展追踪

(续)

命 令	描 述
show all	生成扩展的远程访问诊断报告
show cmtracing	显示连接管理者记录是否激活
show configuration	显示配置信息
show installation	显示安装信息
show loglevel	显示RRAS的全局日志级别
show logs	显示所有日志
show modemtracing	显示网络连接中调制解调器设置与消息追踪是否激活
show rastracing	显示对某组件的扩展追踪是否激活
show securityeventlog	显示安全事件记录是否激活
show tracefacilities	显示对所有组件的扩展追踪是否激活

表B-22 Netsh Ras Ip与Netsh Ras Ipv6上下文中的命令

上下文/命令	描 述
netsh ras ip	
add range	向静态IP地址池中添加一个范围
delete pool	从静态IP地址池中删除所有范围
delete range	从静态IP地址池中删除一个范围
dump	显示上下文内配置脚本设置
set access	设置客户端是否可以访问远程访问服务器
set addrassign	设置远程访问服务器为其客户端分配IP地址的方法
set addrreq	设置客户端是否可以请求自己的IP地址
set broadcastnameresolution	设置是否激活或禁用广播名解析（使用基于TCP/IP的NetBIOS）
set negotiation	设置是否为客户端远程访问连接协商IP
set preferredadapter	指定用于路由与远程访问服务的首选适配器
show config	显示当前的远程访问IP配置
show preferredadapter	显示用于路由与远程访问服务的首选适配器
netsh ras ipv6	
dump	显示上下文内配置脚本设置
set access	设置客户端是否可以访问远程访问服务器
set negotiation	设置是否为客户端远程访问连接协商Ipv6
set prefix	设置RAS服务器使用的前缀
show config	显示当前的远程访问Ipv6配置

Netsh Routing

在将软件路由器配置为路由与远程访问服务（RRAS）的一部分时，Netsh Routing上下文可以用于管理IP路由的配置。表B-23总结了这一上下文及Routing Demanddial子上下文中可用的命令，表B-24总结了Routing Ip子上下文中的命令。

表B-23 Netsh Routing与Netsh Routing Demanddial上下文中的命令

上下文/命令	描 述
netsh routing	
demanddial	切换到netsh routing demanddial上下文
dump	显示上下文内配置脚本设置
Ip	切换到netsh routing ip上下文
ipv6	切换到netsh routing ipv6上下文
reset	将IP路由重置为干净状态
add interface	添加一个新的请求拨号接口
delete interface	删除一个请求拨号接口
set callbackdevice	设置用于请求拨号接口的回拨设备的回拨号
set credentials	设置请求拨号接口的拨号凭据
set interface	设置请求拨号接口的选项设置
set ppp	设置请求拨号接口的PPP选项
set security	设置请求拨号接口的安全选项
show callbackdevice	显示用于请求拨号接口的回拨设备的回拨号
show interface	显示请求拨号接口的设置
show ppp	显示请求拨号接口的PPP选项
show security	显示请求拨号接口的安全选项

表B-24 Netsh Routing与Netsh Routing Ip上下文中的命令

命 令	描 述
add boundary	在接口上添加一个多播范围边界
add filter	在指定的接口上添加一个数据包过滤器
add interface	在接口上激活IP转发
add persistentroute	添加持久性静态路由
add preferenceforprotocol	添加路由协议的首选级别
add rtmroute	添加非持久性（NetMgmt）路由
add scope	添加一个多播范围
autodhcp	切换到netsh routing ip autodhcp上下文
delete boundary	从接口上删除一个多播范围边界
delete filter	从指定的接口上删除一个数据包过滤器
delete interface	在指定的接口上删除IP转发
delete persistentroute	删除持久性静态路由
delete preferenceforprotocol	删除指定协议的首选
delete rtmroute	删除用于网络管理的非持久性路由
delete scope	删除一个多播范围
dnsproxy	切换到netsh routing ip dnsproxy上下文
dump	显示上下文内配置脚本设置
igmp	切换到netsh routing ip igmp上下文
nat	切换到netsh routing ip nat上下文

(续)

命 令	描 述
relay	切换到netsh routing ip relay上下文
reset	将IP路由重置为干净状态
rip	切换到netsh routing ip rip上下文
routerdiscovery	切换到netsh routing ip routerdiscovery上下文
set filter	改变指定接口上的过滤器属性
set interface	设置接口状态
set loglevel	设置全局记录级别
set persistentroute	修改持久性静态路由
set preferenceforprotocol	设置路由协议的首选级别
set rtmroute	修改非持久性 (NetMgmt) 路由
set scope	设置多播范围名
set updaterroutes	更新指定的或所有接口的路由信息
show boundary	显示已配置的多播范围边界
show boundarystats	显示IP多播边界
show filter	显示数据包过滤器信息
show interface	显示接口信息
show loglevel	显示全局记录级别
show mfe	显示多播转发条目
show mfestats	显示多播转发条目统计信息
show persistentroutes	显示持久性静态路由
show preferenceforprotocol	显示所有协议的首选级别
show protocol	显示所有已配置的IP协议
show rtmdestinationations	显示路由表中的目的地址
show rtmroutes	显示路由表中的路由信息
show scope	显示路由器上配置的多播范围
update	更新接口上的自动静态路由

表B-25、表B-26、表B-27、表B-28总结了Netsh Routing Ip上下文的一些子上下文中可用的命令。其中，表B-25总结了Netsh Routing Ip Autodhcp与Netsh Routing Ip Dnsproxy中的命令，表B-26总结了Netsh Routing Ip Igmp与Netsh Routing Ip Nat中的命令，表B-27总结了Netsh Routing Ip Relay与Netsh Routing Ip Rip中的命令，表B-28总结了Netsh Routing Ip Routerdiscovery、Netsh Routing Ipv6、Netsh Routing Ipv6 Relayv6中的命令。

表B-25 Netsh Routing Ip Autodhcp与Netsh Routing Ip Dnsproxy中的命令

上下文/命令	描 述
netsh routing ip autodhcp	
add exclusion	为DHCP分配器范围添加排除
delete exclusion	从DHCP分配器范围中删除排除
dump	显示上下文内配置脚本设置

(续)

上下文/命令	描 述
install	安装与当前上下文相对应的路由协议
set global	修改全局DHCP分配器参数
set interface	修改某接口DHCP分配器参数
show global	显示DHCP分配器配置
show interface	显示指定接口的DHCP分配器配置
uninstall	移除与当前上下文相对应的路由协议
netsh routing ip dnsproxy	
dump	显示上下文内配置脚本设置
install	安装与当前上下文相对应的路由协议
set global	设置全局DNS代理参数
set interface	设置某接口DNS代理参数
show global	显示DNS代理配置
show interface	显示指定接口的DNS代理配置
uninstall	移除与当前上下文相对应的路由协议

表B-26 Netsh Routing Ip Igmp与Netsh Routing Ip Nat中的命令

上下文/命令	描 述
netsh routing ip igmp	
add interface	在指定的接口上配置IGMP
delete interface	从指定的接口上移除IGMP路由器/代理
dump	显示上下文内配置脚本设置
install	安装IGMP路由器/代理并设置全局记录
set global	设置IGMP全局参数
set interface	修改接口配置参数
show global	显示全局IGMP参数
show groupable	显示多播组的IGMP主机组表
show ifstats	显示指定接口的IGMP统计信息
show iftable	显示指定接口的IGMP主机组
show interface	显示接口IGMP信息
show proxygroupable	显示IGMP代理接口的IGMP主机组表
show rasgroupable	显示远程访问客户端接口的主机组表
uninstall	移除与当前上下文相对应的路由协议
netsh routing ip nat	
add addressmapping	向NAT接口地址池中添加IP地址映射
add addressrange	向NAT接口地址池中添加地址范围
add ftp	激活FTP代理
add interface	在指定的接口上配置NAT
add portmapping	在NAT接口上添加协议端口映射
delete addressmapping	从NAT接口地址池中删除IP地址映射

(续)

上下文/命令	描 述
delete addressrange	从NAT接口地址池中删除地址范围
delete ftp	禁用FTP代理
delete interface	从指定的接口上移除NAT
delete portmapping	从NAT接口上移除协议端口映射
dump	显示上下文内配置脚本设置
install	安装与当前上下文相对应的路由协议
set global	设置全局NAT参数
set interface	修改某接口NAT参数
show global	显示NAT配置
show interface	显示指定接口的NAT配置
uninstall	移除与当前上下文相对应的路由协议

表B-27 Netsh Routing Ip Relay与Netsh Routing Ip Rip中的命令

上下文/命令	描 述
netsh routing ip relay	
add dhcpserver	向DHCP服务器全局列表添加一台DHCP服务器
add interface	激活接口上的DHCP Relay agent
delete dhcpserver	从DHCP服务器全局列表删除一台DHCP服务器
delete interface	禁用接口上的DHCP Relay agent
dump	显示上下文内配置脚本设置
install	安装与当前上下文相对应的路由协议
set global	设置DHCP Relay agent配置的全局参数
set interface	更新接口上的DHCP Relay agent配置
show global	显示DHCP Relay agent全局配置
show ifbinding	显示接口的IP地址绑定
show ifconfig	显示每个接口配置的DHCP Relay agent
show ifstats	显示每个接口配置的DHCP Relay agent统计信息
show interface	显示特定接口的DHCP Relay agent配置
uninstall	移除与当前上下文相对应的路由协议
netsh routing ip rip	
add acceptfilter	为某接口上接收的路由添加一个接收过滤器
add announcefilter	为某接口上宣称的路由添加一个过滤器
add interface	在指定的接口上配置RIP
add neighbor	在某接口上添加RIP邻居
add peerfilter	为可以接受为端点的服务器添加过滤器
delete acceptfilter	从某接口上接收的路由中删除一个接收过滤器
delete announcefilter	从某接口上宣称的路由中删除一个过滤器
delete interface	从指定的接口上移除RIP
delete neighbor	从某接口上删除RIP邻居

(续)

上下文/命令	描 述
delete peerfilter	从已接受的端点服务器删除过滤器
dump	显示上下文内配置脚本设置
install	安装与当前上下文相对应的路由协议
set flags	为某指定接口设置RIP相关的标志
set global	设置全局RIP参数
set interface	修改指定端口上的RIP配置
show flags	显示指定接口的RIP标志集
show global	显示RIP全局参数
show globalstats	显示RIP全局统计信息
show ifbinding	显示RIP接口的IP地址绑定
show ifstats	显示每个接口的RIP统计信息
show interface	显示指定接口的RIP配置
show neighbor	显示RIP端点统计信息
uninstall	移除与当前上下文相对应的路由协议

表B-28 Netsh Routing Ip Routerdiscovery、Netsh Routing Ipv6、
Netsh Routing Ipv6 Relayv6中的命令

上下文/命令	描 述
netsh routing ip routerdiscovery	
add interface	为指定接口配置路由器发现
delete interface	从指定接口移除路由器发现
dump	显示上下文内配置脚本设置
set interface	更新接口的路由器发现配置
show interface	显示路由器发现信息
uninstall	移除与当前上下文相对应的路由协议
netsh routing ipv6	
add filter	向指定接口添加IPv6数据包过滤器
add persistentroute	向指定接口添加持久性路由
delete filter	从指定接口删除IPv6数据包过滤器
delete persistentroute	删除持久性静态路由
dump	显示上下文内配置脚本设置
relayv6	切换到netsh routing ipv6 relayv6上下文
set filter	改变指定接口上的IPv6过滤器属性
set persistentroute	修改持久性静态路由
show filter	显示IPv6数据包过滤器信息
show persistentroutes	显示持久性静态路由
netsh routing ipv6 relayv6	
add dhcpserver	向DHCPv6服务器全局列表添加DHCPv6服务器
add interface	激活接口上的DHCPv6 Relay Agent

(续)

上下文/命令	描 述
delete dhcpserver	从DHCPv6服务器全局列表中删除DHCPv6服务器
delete interface	禁用接口上的DHCPv6 Relay Agent
dump	显示上下文内配置脚本设置
install	安装与当前上下文相对应的路由协议
set global	设置DHCPv6 Relay Agent的全局参数
set interface	更新接口上的DHCPv6 Relay Agent配置
show global	显示DHCPv6 Relay Agent全局配置
show interface	显示特定端口的DHCPv6 Relay Agent配置
uninstall	移除与当前上下文相对应的路由协议

Netsh Rpc

Netsh Rpc上下文与Netsh Rpc Filter上下文可以用于操作远程过程调用（RPC）防火墙，表B-29总结了这些上下文中可用的命令。

表B-29 Netsh Rpc与Netsh Rpc Filter上下文中的命令

上下文/命令	描 述
netsh rpc	
add	创建一个子网添加列表
delete	创建一个子网删除列表
dump	显示上下文内配置脚本设置
filter	切换到netsh rpc filter上下文
reset	将选择性的绑定设置重置为“空”（在所有接口上监听）
show	显示系统上每一子网的选择性绑定状态
netsh rpc filter	
add condition	向现有的RPC防火墙过滤器规则中添加条件
add filter	添加一个RPC防火墙过滤器
add rule	添加一条RPC防火墙过滤器规则
delete filter	删除一个RPC防火墙过滤器
delete rule	添加一条RPC防火墙过滤器规则
dump	显示上下文内配置脚本设置
show filter	列出RPC防火墙过滤器

Netsh Winhttp

Netsh Winhttp上下文可以用于管理Windows HTTP（WinHTTP）代理与追踪设置，该上下文中可用的命令包括下面列举的。

- **dump**。显示配置脚本

- ❑ **import**。导入WinHTTP代理设置。
- ❑ **reset proxy**。将WinHTTP代理设置重置为直接值。
- ❑ **reset tracing**。将WinHTTP追踪参数重置为其默认值。
- ❑ **set proxy**。配置WinHTTP代理设置。
- ❑ **set tracing**。配置WinHTTP追踪参数。
- ❑ **show proxy**。显示当前WinHTTP代理设置。
- ❑ **show tracing**。显示当前WinHTTP追踪参数。

Netsh Wins

Netsh Wins上下文与Netsh Wins Server上下文可以用于管理Windows Internet命名服务（WINS）服务器及其配置。表B-30总结了这些上下文中可用的命令。要注意的是，只有当Windows Server计算机上已经安装了WINS Server功能时，这些上下文才是可用的。如果尚未安装该功能而使用Wins命令，则会导致进入Winsock上下文。

表B-30 Netsh Wins上下文与Netsh Wins Server上下文中的命令

上下文/命令	描 述
netsh wins	
dump	显示配置
server[\\ServerName \\IPAddress]	切换到指定的服务器上下文。如果不指定服务器名或IP地址，则默认操作的是本地计算机上的WINS服务器
netsh wins server	
add name	向服务器中添加名称记录
add partner	向服务器中添加复制参与方
add pgserver	向当前服务器中添加Persona Grata Servers列表
add pngserver	向当前服务器中添加Persona Non Grata Servers列表
check database	检查数据库的一致性
check name	检查WINS服务器集的名称记录列表
check version	检查版本号的一致性
delete name	从服务器数据库中删除注册名
delete owners	删除属主列表及其记录
delete partner	从复制参与方列表中删除一个复制参与方
delete pgserver	从列表中删除所有的或选定的Persona Grata Servers
delete pngserver	从列表中删除所有的或选定的Persona Non Grata Servers
delete records	从服务器中删除或冻结所有的或部分记录集
dump	显示配置
init backup	初始化WINS数据库的备份
init import	初始化从LMHOSTS文件的导入
init pull	初始化并将pull触发器发送到另外的WINS服务器
init pullrange	初始化并从另外的WINS服务器复制选定反问的记录
init push	初始化并将push触发器发送到另外的WINS服务器
init replicate	初始化与复制参与方的数据库复制

(续)

上下文/命令	描 述
init restore	初始化从文件进行的数据库恢复
init scavenge	初始化清除服务器的WINS数据库
init search	初始化在WINS服务器数据库中搜索（用来发现指定的记录）
reset	重置表中的配置条目
set autopartnerconfig	为服务器设置自动化的复制参与方配置信息
set backuppath	为服务器设置备份参数
set burstparam	为服务器设置紧急处理参数
set defaultparam	将WINS Server配置参数设置为默认值
set logparam	设置数据库与事件记录选项
set migrateflag	设置服务器的迁移标记
set namerecord	设置服务器的间隔与超时值
set periodicdbchecking	设置服务器的周期性数据库检测参数
set pgmode	设置Persona Grata/ Non Grata模式
set pullparam	设置服务器的默认pull参与方参数
set pullpartnerconfig	设置指定pull参与方的配置参数
set pushparam	设置服务器的默认push参与方参数
set pushpartnerconfig	设置指定push参与方的配置参数
set replicateflag	设置服务器的复制标记
set startversion	设置数据库的开始版本ID
show browser	显示所有活跃域主机浏览器[1Bh]记录
show database	为所有或部分指定属主服务器显示数据库与记录
show info	显示服务器配置信息
show name	显示服务器中特定记录的详细信息
show partner	显示服务器的所有参与方，或push参与方，或pull参与方
show partnerproperties	显示默认的参与方配置
show pullpartnerconfig	显示pull参与方的配置信息
show pushpartnerconfig	显示push参与方的配置信息
show recbyversion	显示特定服务器拥有的记录
show reccount	显示特定属主服务器拥有的记录数目
show server	显示当前选定的服务器
show statistics	显示WINS服务器的统计信息
show version	显示WINS服务器的当前最大版本计数器值
show versionmap	显示属主ID到Maximum Version Number的映射

Netsh Winsock

Netsh Winsock上下文可以用于管理Winsock通信，该上下文中可用的命令包括下面列举的。

- ❑ **audit trial**。展示分层服务提供商（LSP）安装与卸载的审计踪迹。
- ❑ **dump**。显示配置脚本。

- ❑ **remove provider**。从系统中移除Winsock LSP。
- ❑ **reset**。将Winsock Catalog重置为干净状态。
- ❑ **show catalog**。显示Winsock Catalog的内容。

Netsh Wlan

Netsh Wlan上下文可以用于管理计算机的无线网络配置，表B-31总结了该上下文中可用的命令。

表B-31 Netsh Wlan上下文中的命令

命 令	描 述
add filter	将无线网络添加到允许或阻止的无线网络列表
add profile	将WLAN配置文件添加到系统中指定的接口上
connect	连接到无线网络
delete filter	从允许或阻止的无线网络列表中删除无线网络
delete profile	从一个或多个接口上删除WLAN配置文件
disconnect	断开与无线网络的连接
dump	显示上下文内配置脚本设置
export	将WLAN配置文件保存到XML文件
set autoconfig	激活或禁用接口上的自动配置逻辑
set blockednetworks	在可视网络列表中显示或隐藏阻止的网络
set createalluserprofile	允许或禁止每个人创建所有用户的配置文件
set profileorder	设置无线网络配置文件的首选顺序
set tracing	激活或禁用追踪
show all	显示完整的无线设备与网络信息
show autoconfig	显示自动配置逻辑处于激活还是禁用状态
show blockednetworks	显示阻止的网络显示设置
show createalluserprofile	显示是否每个人都可以创建所有用户配置文件
show drivers	显示系统上无线LAN驱动程序的属性
show filters	显示允许或阻止的网络列表
show interfaces	显示系统上无线LAN接口列表
show networks	显示系统上可见网络列表
show profiles	显示系统上已配置的配置文件列表
show settings	显示无线LAN的全局设置
show tracing	显示无线LAN追踪是激活还是禁用

[G e n e r a l I n f o r m a t i o n]

书名=WINDOWS命令行详解手册

作者=(美)WilliamR.Stanek著

页数=400

出版社=北京市：人民邮电出版社

出版日期=2009.08

SS号=12336379

DX号=000006772877

URL=<http://book.szdnet.org.cn/bookDetail.jsp?dxNumber=000006772877&d=0E4050F24D2D832D07D8DECA3F76C2F6>

第一部分 Windows 命令行基础第 1 章 Windows 命令行概述

- 1.1 命令行基础
 - 1.1.1 理解 Windows 命令 shell
 - 1.1.2 理解 MS-DOS 命令 shell
 - 1.1.3 理解 Windows PowerShell
 - 1.1.4 配置命令行属性
 - 1.1.5 使用命令历史
- 1.2 使用补充的组件
 - 1.2.1 在 Windows Vista 中使用微软远程服务器管理工具
 - 1.2.2 注册远程服务器管理工具包
 - 1.2.3 配置与选择远程服务器管理工具
 - 1.2.4 删除远程服务器管理工具
 - 1.2.5 删除远程服务器管理工具软件包

第 2 章 充分利用命令行

- 2.1 管理命令 shell 的启动方式
- 2.2 使用命令路径进行工作
 - 2.2.1 管理命令路径
 - 2.2.2 管理文件扩展与文件关联
- 2.3 标准输入、输出及错误日志的重定向
 - 2.3.1 将标准输出重定向到其他命令
 - 2.3.2 I/O 与文件的重定向
 - 2.3.3 标准错误输出的重定向
- 2.4 命令的结链与分组
 - 2.4.1 使用命令链
 - 2.4.2 命令分组

第 3 章 命令行脚本基础

- 3.1 创建命令行脚本
- 3.2 脚本的常见语句与命令
 - 3.2.1 清除命令 shell 窗口
 - 3.2.2 为脚本添加注释
 - 3.2.3 管理文字的显示方式与命令回显方式
 - 3.2.4 使用 @ 对命令回显进行调整
 - 3.2.5 设置控制台窗口的标题与颜色
- 3.3 向脚本传递参数
- 3.4 熟悉变量
- 3.5 在脚本中使用变量
 - 3.5.1 变量命名
 - 3.5.2 设置变量值
 - 3.5.3 替换变量值
 - 3.5.4 变量作用范围局部化
- 3.6 使用数学表达式
 - 3.6.1 使用算术运算符与赋值运算符
 - 3.6.2 理解运算符的优先级

3.6.3	模拟指数操作	
3.7	命令行选择语句	
3.7.1	使用 if 语句	
3.7.2	使用 if not 语句	
3.7.3	使用 if defined 与 if not defined 语句	
3.7.4	使用嵌套的 if 语句	
3.7.5	在 if 语句中进行比较	
3.8	命令行迭代语句	
3.8.1	迭代的基础	
3.8.2	遍历一系列值	
3.8.3	在成组的文件中迭代执行	
3.8.4	在目录中迭代执行	
3.8.5	分析文件的内容与输出	
3.9	创建子程序与过程	
3.9.1	使用子程序	
3.9.2	使用过程	
第二部分	使用命令行管理 Windows 系统	第 4 章 部署 Windows 服务器
4.1	服务器配置管理	
4.2	使用角色、角色服务与功能	
4.3	管理角色、角色服务与功能	
4.3.1	Server Manager Cmd 基础	
4.3.2	查询已安装的角色、角色服务与功能	
4.3.3	安装角色、角色服务与功能	
4.3.4	移除角色、角色服务与功能	
第 5 章	管理 Windows 系统	
5.1	检查系统信息	
5.2	操作注册表	
5.2.1	理解注册表与键值	
5.2.2	查询注册表值	
5.2.3	比较注册表值	
5.2.4	注册表键的保存与恢复	
5.2.5	添加注册表键	
5.2.6	复制注册表键	
5.2.7	删除注册表键	
5.2.8	导入与导出注册表键	
5.2.9	加载与卸载注册表键	
5.3	管理系统服务	
5.3.1	查看已配置的服务	
5.3.2	启动、终止与暂停服务	
5.3.3	配置服务的启动方式	
5.3.4	配置服务的登录方式	
5.3.5	配置服务的恢复方式	
5.4	从命令行重启与关闭系统	
5.4.1	管理本地系统的重启与关闭	
5.4.2	管理远程系统的重启与关闭	
5.4.3	添加关机或重启原因与注释	
第 6 章	事件记录、追踪与监控	
6.1	Windows 事件日志	

- 6 . 2 查看与过滤事件日志
 - 6 . 2 . 1 查看事件
 - 6 . 2 . 2 过滤事件
- 6 . 3 向事件日志中写入自定义事件
- 6 . 4 创建与使用保存的查询
- 6 . 5 性能监控：基础
 - 6 . 5 . 1 理解如何在命令行中进行性能监控
 - 6 . 5 . 2 追踪性能数据
- 第 7 章 进程监控与性能维护
 - 7 . 1 管理应用程序、进程与性能
 - 7 . 1 . 1 理解系统与用户进程
 - 7 . 1 . 2 检查运行中进程
 - 7 . 1 . 3 监控系统资源使用情况与进程
 - 7 . 1 . 4 终止进程
 - 7 . 2 通过监控来检测与解决性能问题
 - 7 . 2 . 1 监控内存分页与磁盘页面
 - 7 . 2 . 2 监控单个进程的内存使用与Working Memory Set
 - 7 . 2 . 3 解决性能瓶颈
- 第 8 章 管理事件与性能日志
 - 8 . 1 管理事件日志
 - 8 . 1 . 1 开始使用Wevtutil
 - 8 . 1 . 2 列出可用的日志与已注册的事件发布者
 - 8 . 1 . 3 查看与改变日志配置
 - 8 . 1 . 4 导出与操作事件日志
 - 8 . 1 . 5 清除事件日志
 - 8 . 2 企业级集中化事件记录机制
 - 8 . 2 . 1 配置事件转发与收集
 - 8 . 2 . 2 创建订阅
 - 8 . 2 . 3 管理订阅
 - 8 . 3 性能日志
 - 8 . 3 . 1 开始使用数据收集器集
 - 8 . 3 . 2 操作数据收集器集
 - 8 . 3 . 3 收集性能计数器数据
 - 8 . 3 . 4 配置性能计数器警报
 - 8 . 3 . 5 查看数据收集器报告
- 第 9 章 计划任务的自动运行
 - 9 . 1 在本地与远程系统上执行计划任务
 - 9 . 1 . 1 计划任务简介
 - 9 . 1 . 2 监控计划任务
 - 9 . 2 使用任务计划程序计划任务
 - 9 . 2 . 1 创建基本任务
 - 9 . 2 . 2 创建高级任务
 - 9 . 2 . 3 管理任务属性
 - 9 . 2 . 4 激活与禁用任务
 - 9 . 2 . 5 将任务复制到其他计算机
 - 9 . 2 . 6 立即运行任务
 - 9 . 2 . 7 移除不需要的任务
 - 9 . 3 使用Schtasks设置任务计划

9.3.1	使用 S c h t a s k s / C r e a t e 创建计划任务	
9.3.2	创建由 W i n d o w s 事件触发的计划任务	
9.3.3	使用 S c h t a s k s / C h a n g e 改变计划任务	
9.3.4	使用 S c h t a s k s / Q u e r y 查询已配置的任务	
9.3.5	使用 X M L 配置文件创建任务	
9.3.6	使用 S c h t a s k s / R u n 立即运行任务	
9.3.7	使用 S c h t a s k s / E n d 终止运行中的任务	
9.3.8	使用 S c h t a s k s / D e l e t e 删除任务	
第三部分	使用命令行管理 W i n d o w s 文件系统和磁盘第 10 章	配置与维护磁盘
10.1	使用 D i s k P a r t	
10.1.1	D i s k P a r t 基础	
10.1.2	D i s k P a r t : 一个实例	
10.1.3	理解焦点及其内涵	
10.1.4	D i s k P a r t 命令与脚本	
10.1.5	D i s k P a r t : 脚本实例	
10.2	安装与管理硬盘驱动器	
10.2.1	安装与检查新驱动器	
10.2.2	检查驱动器状态与配置	
10.2.3	修改驱动器分区风格	
10.3	操作基本磁盘与动态磁盘	
10.3.1	理解基本磁盘与动态磁盘	
10.3.2	设置活动分区	
10.3.3	改变磁盘类型: 基本磁盘与动态磁盘的互相转换	
10.4	磁盘维护	
10.4.1	使用 F S U t i l 获取磁盘信息并管理文件系统	
10.4.2	检查磁盘的错误与坏扇区	
10.4.3	修正磁盘错误	
10.4.4	对系统启动时的自动检测进行控制	
10.5	磁盘碎片整理	
第 11 章	对基本磁盘进行分区	
11.1	获取分区信息	
11.2	创建分区	
11.2.1	在 M B R 磁盘上创建分区	
11.2.2	在 G P T 磁盘上创建分区	
11.3	管理盘符与挂载点	
11.3.1	分配驱动器盘符或挂载点	
11.3.2	改变驱动器盘符或挂载点	
11.3.3	移除盘符或挂载点	
11.4	格式化分区	
11.4.1	使用 F O R M A T	
11.4.2	使用 F I L E S Y S T E M S	
11.4.3	格式化: 一个实例	
11.5	管理分区	
11.5.1	将分区或卷转换为 N T F S	
11.5.2	改变或删除卷标	
11.5.3	压缩分区或卷	
11.5.4	扩展分区或卷	
11.5.5	删除分区	

第 1 2 章 管理动态磁盘上的卷与 R A I D

- 1 2 . 1 获取卷信息与状态
- 1 2 . 2 创建并管理简单卷
 - 1 2 . 2 . 1 创建简单卷
 - 1 2 . 2 . 2 扩展简单卷
 - 1 2 . 2 . 3 将动态磁盘联机
 - 1 2 . 2 . 4 删除卷
- 1 2 . 3 通过动态磁盘上的 R A I D 提供容错功能
 - 1 2 . 3 . 1 实现 R A I D - 0 : 磁盘分割
 - 1 2 . 3 . 2 实现 R A I D - 1 : 磁盘镜像与双控
 - 1 2 . 3 . 3 实现 R A I D - 5 : 带奇偶校验的磁盘分割
- 1 2 . 4 管理 R A I D 并从失效中恢复
 - 1 2 . 4 . 1 分离镜像集
 - 1 2 . 4 . 2 重新同步与修复镜像集
 - 1 2 . 4 . 3 修复不带奇偶校验信息的 R A I D - 0 条带集
 - 1 2 . 4 . 4 重建带奇偶校验信息的 R A I D - 5 条带集

第四部分 使用命令行管理 W i n d o w s 活动目录第 1 3 章 核心目录服务管理

- 1 3 . 1 从命令行控制活动目录
 - 1 3 . 1 . 1 理解域、容器与对象
 - 1 3 . 1 . 2 理解活动目录中的逻辑结构与物理结构
 - 1 3 . 1 . 3 理解区分名
 - 1 3 . 1 . 4 使用活动目录命令行工具
- 1 3 . 2 使用 D S Q U E R Y 命令进行目录查询
 - 1 3 . 2 . 1 D S Q U E R Y 子命令及语法
 - 1 3 . 2 . 2 使用名称、描述、S A M 账号名进行搜索
 - 1 3 . 2 . 3 设定搜索的登录域与 R u n A s 许可权限
 - 1 3 . 2 . 4 设定开始节点、搜索范围与对象限制
 - 1 3 . 2 . 5 设定名的输出格式
 - 1 3 . 2 . 6 结合使用 D S Q U E R Y 与其他活动目录命令行工具
- 1 3 . 3 搜索问题用户与计算机账号
- 1 3 . 4 对象的重命名与移动
- 1 3 . 5 从活动目录中移除对象

第 1 4 章 管理计算机账号与域控制器

- 1 4 . 1 从命令行管理计算机账号概览
- 1 4 . 2 在活动目录域内创建计算机账号
 - 1 4 . 2 . 1 创建计算机账号
 - 1 4 . 2 . 2 定制计算机账号属性与组成员关系
- 1 4 . 3 管理计算机账号属性
 - 1 4 . 3 . 1 查看与寻找计算机账号
 - 1 4 . 3 . 2 设置或修改计算机的位置与描述信息属性
 - 1 4 . 3 . 3 禁用与激活计算机账号
 - 1 4 . 3 . 4 重置锁定的计算机账号
 - 1 4 . 3 . 5 将计算机账号添加到某域中
 - 1 4 . 3 . 6 对计算机与计算机账号进行重命名
 - 1 4 . 3 . 7 移动计算机账号
 - 1 4 . 3 . 8 删除计算机账号
- 1 4 . 4 操作域控制器
 - 1 4 . 4 . 1 安装与降级域控制器

1 4 . 4 . 2	在活动目录中发现域控制器
1 4 . 5	指定全局编目服务器
1 4 . 5 . 1	发现全局编目服务器
1 4 . 5 . 2	添加或移除全局编目服务器
1 4 . 5 . 3	检查缓存与优先的全局编目设置
1 4 . 6	指定操作主机
1 4 . 6 . 1	发现操作主机
1 4 . 6 . 2	使用命令行配置操作主机角色
1 4 . 7	发现只读的域控制器
第 1 5 章	管理活动目录用户与组
1 5 . 1	从命令行中管理用户账号概览
1 5 . 2	添加用户账号
1 5 . 2 . 1	创建域用户账号
1 5 . 2 . 2	自定义域用户账号属性与组成员关系
1 5 . 2 . 3	创建本地用户账号
1 5 . 3	管理用户账号
1 5 . 3 . 1	查看与查找用户账号
1 5 . 3 . 2	确定单独用户账号的组成员关系
1 5 . 3 . 3	设置或更改用户账号属性
1 5 . 3 . 4	禁用与激活用户账号
1 5 . 3 . 5	重置过期的用户账号
1 5 . 3 . 6	控制与重置用户口令
1 5 . 3 . 7	移动用户账号
1 5 . 3 . 8	用户账号重命名
1 5 . 3 . 9	删除用户账号
1 5 . 4	从命令行管理组账号概览
1 5 . 5	添加组账号
1 5 . 5 . 1	创建安全组与分发组
1 5 . 5 . 2	创建本地组并为其分配成员
1 5 . 6	管理组账号
1 5 . 6 . 1	查看与寻找组账号
1 5 . 6 . 2	确定组成员关系
1 5 . 6 . 3	改变组类型或范围
1 5 . 6 . 4	添加、移除或替换组成员
1 5 . 6 . 5	移动组账号
1 5 . 6 . 6	组账号重命名
1 5 . 6 . 7	删除组账号
第五部分	使用命令行管理网络
第 1 6 章	管理网络打印机与打印服务
1 6 . 1	获取打印机的支持信息与故障排除信息
1 6 . 1 . 1	在命令行中操作打印机
1 6 . 1 . 2	追踪打印驱动程序与打印机信息
1 6 . 1 . 3	获取用于容量规划与故障排除的打印详细统计资料
1 6 . 2	管理打印机
1 6 . 2 . 1	安装物理连接的打印设备
1 6 . 2 . 2	安装网络连接的打印设备
1 6 . 2 . 3	列出计算机上配置的打印机
1 6 . 2 . 4	查看与设置默认打印机
1 6 . 2 . 5	打印机重命名

1 6 . 2 . 6	删除打印机
1 6 . 3	管理网络连接打印机的 T C P / I P 端口
1 6 . 3 . 1	为打印机创建与改变 T C P / I P 端口
1 6 . 3 . 2	列出打印机使用的 T C P / I P 端口相关的信息
1 6 . 3 . 3	删除打印机使用的 T C P / I P 端口
1 6 . 4	配置打印机属性
1 6 . 4 . 1	添加注释与位置信息
1 6 . 4 . 2	共享打印机
1 6 . 4 . 3	在活动目录中发布打印机
1 6 . 4 . 4	设置分隔页并改变打印设备模式
1 6 . 4 . 5	打印任务的调度与优先级设置
1 6 . 4 . 6	配置缓冲池与其他高级打印机选项
1 6 . 5	解决缓存问题
1 6 . 5 . 1	检查 P r i n t S p o o l e r 服务
1 6 . 5 . 2	修复损坏的缓冲池
1 6 . 6	管理打印队列与单个打印任务
1 6 . 6 . 1	查看队列中的任务
1 6 . 6 . 2	打印机的暂停与恢复
1 6 . 6 . 3	清空打印队列
1 6 . 6 . 4	暂停、恢复与重启单个文档的打印
1 6 . 6 . 5	移除文档并取消打印任务
1 6 . 7	备份与恢复打印服务器配置
1 6 . 7 . 1	备份打印服务器的配置
1 6 . 7 . 2	恢复打印服务器的配置
1 6 . 7 . 3	迁移打印机与打印队列
第 1 7 章	T C P / I P 网络的配置、管理与故障排除
1 7 . 1	使用网络服务 S h e l l
1 7 . 1 . 1	操作 N e t s h 上下文
1 7 . 1 . 2	操作远程计算机
1 7 . 1 . 3	操作脚本文件
1 7 . 2	管理 T C P / I P 设置
1 7 . 2 . 1	配置 I P v 4
1 7 . 2 . 2	配置 I P v 6
1 7 . 3	支持 T C P / I P 网络
1 7 . 3 . 1	获取并保存 T C P / I P 设置
1 7 . 3 . 2	检查 I P 地址与网络接口配置
1 7 . 3 . 3	操作 T C P I n t e r n e t 控制与错误消息
1 7 . 3 . 4	检查分片、重组、错误消息的详细信息
1 7 . 3 . 5	检查当前的 T C P 与 U D P 连接
1 7 . 4	排除 T C P / I P 网络故障
1 7 . 4 . 1	查看诊断信息
1 7 . 4 . 2	诊断常规的计算机配置问题
附录 A	基本命令行工具参考
附录 B	N e t s h 快速参考